



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ

FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY

INSTITUTE OF INFORMATICS

NÁVRH DATABÁZE PRO VYBRANOU SPOLEČNOST

DESIGN OF DATABASE FOR SELECTED COMPANY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Tomáš Novák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jan Luhan, Ph.D., MSc

BRNO 2017

Zadání bakalářské práce

Ústav: Ústav informatiky
Student: **Tomáš Novák**
Studijní program: Systémové inženýrství a informatika
Studijní obor: Manažerská informatika
Vedoucí práce: **Ing. Jan Luhan, Ph.D., MSc**
Akademický rok: 2016/17

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává bakalářskou práci s názvem:

Návrh databáze pro vybranou společnost

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Cílem práce je vytvoření návrhu databáze pro potřeby konkrétního subjektu se zaměřením na jeho obchodní činnost pro zajištění efektivity jeho procesů. Výstupem práce je realizovaný datový návrh na zvolené databázové platformě.

Základní literární prameny:

BASL, J. a R. BLAŽÍČEK. Podnikové informační systémy - podnik v informační společnosti. 3. vyd. Praha: Grada, 2012. 328 s. ISBN 978-80-247-4307-3.

CONOLLY, T., C. E. BEGG a R. HOLOWCZAK. Mistrovství – databáze: Profesionální průvodce tvorbou efektivních databází. 1. vyd. Brno: Computer Press, 2009. 584 s. ISBN 978-80-251-2328-7.

KROENKE, D., D. J. AUER a J. GONER. Databáze. 1. vyd. Brno: Computer Press, 2015. 496 s. ISBN 978-80-251-4352-0.

LACKO, L. 1001 tipů a triků pro SQL. 1. vyd. Brno: Computer Press, 2011. 416 s. ISBN 978-80-251-3010-0.

LACKO, L. Mistrovství v Microsoft SQL Server 2012. 1. vyd. Praha: Computer Press, 2013. 640 s. ISBN 978-80-251-3773-4.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2016/17.

V Brně, dne 28. 2. 2017



doc. RNDr. Bedřich Půža, CSc.
ředitel



doc. Ing. et Ing. Stanislav Škapa, Ph.D.
děkan

Abstrakt

Tato bakalářská práce se zaměřuje na návrh databázového systému pro vybranou společnost, který má zefektivnit vnitřní procesy ve společnosti. První část nastiňuje teoretické poznatky týkající se této práce. Dále obsahuje analýzu společnosti, pro níž je systém navržen a nakonec návrh celého systému.

Abstract

This bachelor thesis focuses on the design of the database system for the selected company, which is to be streamline inner processes in the company. The first part outlines theoretical background regarding this thesis. Further contains analysis of the company for which the system is designed and last proposal of the whole system.

Klíčová slova

databáze, data, normalizace, SQL, Microsoft SQL Server, servis, ER diagram, návrh, relace

Key words

database, data, normalization, SQL, Microsoft SQL Server, service, ER diagram, proposal, relation

Bibliografická citace

NOVÁK, T. *Návrh databáze pro vybranou společnost*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2017. 83 s. Vedoucí bakalářské práce Ing. Jan Luhan, Ph.D., MSc.

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 26. května 2017



podpis studenta

Poděkování

Tímto bych rád poděkoval panu Ing. Janu Luhanovi, Ph.D., MSc za odborné rady a připomínky i čas, který mne věnoval během této práce. Dále bych rád poděkoval panu Danielu Dvořákovi a ostatním zaměstnancům ze společnosti DVOŘÁK TRUCKS za poskytnutí cenných informací k této práci.

OBSAH

| | |
|--|----|
| ÚVOD..... | 9 |
| 1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ..... | 10 |
| 2 TEORETICKÉ VÝCHODISKÁ PRÁCE | 11 |
| 2.1 Data a informace | 11 |
| 2.2 Efekt informačního systému v podniku | 12 |
| 2.3 Databáze..... | 12 |
| 2.3.1 Systém řízení databáze DBMS | 13 |
| 2.3.2 Databázová aplikace | 13 |
| 2.3.3 Architektury DBMS..... | 14 |
| 2.4 Databázové modely..... | 14 |
| 2.4.1 Lineární model..... | 14 |
| 2.4.2 Relační model | 15 |
| 2.4.3 Objektový model..... | 15 |
| 2.5 Relační datový model..... | 15 |
| 2.5.1 Relace..... | 16 |
| 2.5.2 Integritní omezení pro entity..... | 16 |
| 2.5.3 Typy klíčů | 17 |
| 2.5.4 Integritní omezení pro vztahy entit..... | 17 |
| 2.6 Návrh databáze..... | 18 |
| 2.6.1 Konceptuální návrh..... | 18 |

| | | |
|-------|---------------------------------|----|
| 2.6.2 | Logický návrh | 19 |
| 2.6.3 | Fyzický návrh | 19 |
| 2.7 | Normalizace | 20 |
| 2.7.1 | První normální forma..... | 20 |
| 2.7.2 | Druhá normální forma | 21 |
| 2.7.3 | Třetí normální forma..... | 21 |
| 2.8 | Jazyk SQL | 21 |
| 2.8.1 | Datové typy..... | 21 |
| 2.9 | Microsoft SQL Server 2012 | 22 |
| 2.10 | SWOT analýza..... | 23 |
| 3 | ANALÝZA SOUČASNÉHO STAVU | 24 |
| 3.1 | Představení společnosti | 24 |
| 3.2 | Organizační struktura | 25 |
| 3.3 | Konkurence | 26 |
| 3.4 | Informační technologie | 26 |
| 3.4.1 | Hardware..... | 26 |
| 3.4.2 | Software | 27 |
| 3.5 | SWOT analýza | 27 |
| 3.5.1 | Silné stránky | 28 |
| 3.5.2 | Slabé stránky..... | 28 |
| 3.5.3 | Příležitosti | 29 |

| | | |
|-------|---|----|
| 3.5.4 | Hrozby | 29 |
| 3.6 | Současný proces zpracování dat..... | 29 |
| 3.6.1 | Evidence smluv | 29 |
| 3.6.2 | Příjem vozidel na servis a mycí linku | 30 |
| 3.6.3 | Předání vozidla zpět zákazníkovi | 33 |
| 3.7 | Shrnutí situace ve firmě | 33 |
| 4 | VLASTNÍ NÁVRHY ŘEŠENÍ | 35 |
| 4.1 | Požadavky na databázi | 35 |
| 4.2 | Interní procesy..... | 36 |
| 4.2.1 | Vložení nového zákazníka | 36 |
| 4.2.2 | Vytvoření kupní smlouvy | 37 |
| 4.2.3 | Příjem vozidla na servis a mycí linku | 38 |
| 4.2.4 | Sekce faktur | 40 |
| 4.2.5 | Vytvoření smluvního vztahu..... | 40 |
| 4.3 | Konceptuální návrh databáze | 41 |
| 4.4 | Logický návrh databáze | 42 |
| 4.4.1 | Sekce osob | 44 |
| 4.4.2 | Sekce faktur | 45 |
| 4.4.3 | Sekce prodeje | 48 |
| 4.4.4 | Sekce servisu a mycí linky..... | 49 |
| 4.5 | Fyzický návrh databáze | 52 |

| | | |
|--|---|----|
| 4.5.1 | Trigger pro vyskladnění vozidla | 52 |
| 4.5.2 | Procedury na vystavení faktury | 52 |
| 4.6 | Řešení bezpečnosti | 55 |
| 4.6.1 | Rozdělení přístupových údajů | 55 |
| 4.6.2 | Rozdělení zaměstnaneckých rolí | 55 |
| 4.7 | Budoucí uživatelské rozhraní | 56 |
| 4.8 | Předpokládané přínosy návrhu pro podnik | 57 |
| ZÁVĚR | | 58 |
| SEZNAM POUŽITÝCH ZDROJŮ | | 59 |
| SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ | | 61 |
| SEZNAM OBRÁZKŮ | | 62 |
| SEZNAM TABULEK | | 63 |
| SEZNAM PŘÍLOH | | 65 |

ÚVOD

V dnešní době jsou počítačové technologie celosvětově rozšířeny, které se využívají snad ke všem aktivitám podnikání. Jsou však organizace, které informační technologie používají jen zřídka nebo nevyužívají jejich potenciál, to znamená, že jsou nejčastěji využívány k takovým procesům, jako je sběr informací nebo jiné činnosti. Myslím tím, že většina menších firem nedokáže stoprocentně využít či dokonce vůbec obstarat tyto technologie z hlediska financí. Tyto systémy jsou příliš drahé, a proto se takové firmy musí obejít bez nich, což vede k tomu, že v organizaci, které chybí ucelený informační systém, může dojít ke ztrátám nebo zneužití dat.

Z tohoto důvodu je nutné se zaměřit na takové organizace, které by mohli využít informační systémy, ale nemohou si je dovolit. Jsou také organizace, které by tyto systémy nemohli využít v plném rozsahu, z důvodu nenaplnění požadavků.

Proto se tato práce zaměřuje na návrh databázového systému pro firmu, které chybí dostatečný systém a pokryje tak vstupní požadavky organizace z hlediska účelu a ceny.

V následující kapitole jsou vymezeny problémy týkající se vybrané společnosti a cíle, kterými se práce bude zabývat. Další kapitolou jsou teoretické poznatky potřebné k této práci. V analýze společnosti jsou detailně poznamenány nedostatky společnosti a na závěr je zde samotný návrh systému.

1 CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Cílem této práce tedy bude navrhnout databázový systém, který těmto nedostatkům předejde. V této databázi se budou evidovat údaje o zaměstnancích, klientech a dodavatelích. Stěžejním úkolem bude navrhnout systém tak, aby zaznamenával obchodní procesy týkající se prodeje vozidel a vedl o něm náležité účetní doklady. Dále bude vést smluvní vztahy se zákazníky, které zahrnují určité slevy, které je možno využít například v servisu, což je další cíl. Systém musí být navržen tak, aby byl schopen komunikovat se servisním centrem, kde se opět povedou údaje o zákaznících a o proběhlých službách danému zákazníkovi. O těchto službách se povedou opět účetní doklady, jako jsou faktury.

Prvním postupem bylo vytvoření a zhodnocení souhrnné analýzy daného podniku na základně zjištění možných problémů podniku a procesů, které se v tomto subjektu odehrávají. Analýza je pak shrnuta pomocí podkapitol SWOT analýza a Shrnutí situace ve firmě.

V praktické části jsem nejprve začal popisovat interní procesy firmy znázorněné metodou vývojových diagramů. Následně jsem přešel k samotnému návrhu databáze, který je vyjádřen formou 3 úrovní a to konceptuální, kde se zjišťují potřebné entity a vztahy mezi nimi, logická úroveň, ve které došlo k tvorbě jednotlivých tabulek spolu s integritními omezeními a nakonec fyzická úroveň, ve které už návrh dostával podobu požadované databáze. K vytvoření této úrovně byl použit program MS SQL Server 2014, v němž byly strukturovány veškeré tabulky a procedury.

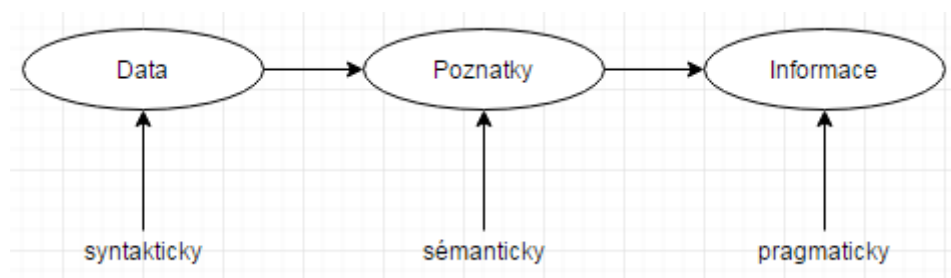
2 TEORETICKÉ VÝCHODISKÁ PRÁCE

Abychom mohli přejít k analýze a návrhovému řešení, musíme nejprve porozumět základním pojmům dané problematiky. Konkrétně nás zajímají databázové systémy, které nás seznámí s pojmy, jako jsou databáze a k čemu nám slouží a také jak postupovat v jednotlivých fázích celého návrhu. Dále vyličím databázový model, který budu používat při návrhu. Nakonec zde bude popsáno prostředí, ve kterém se celý návrh bude vyvíjet.

2.1 Data a informace

Pojem data je v počítačové terminologii používán jako označení pro čísla, text, obraz a jiné vjemy v takové podobě, ve které jsou pokládány jako surová data, jenž jsou vhodné pro počítačové zpracování. Data rozdělujeme na strukturovaná a nestrukturovaná data. Strukturovaná data jsou převážně fakta ukládána do relačních databázových systémů, díky tomu jsme schopni vyvolávat, která požadujeme. Naopak nestrukturovaná data jsou videozáznamy či zvukové nahrávky, která jsou označována jako „*tok bytů*“ (1, s. 2).

Na obr. 1 můžeme vidět jeden z možných způsobů získání informace, nejprve máme v databázi syntakticky uložená data, ale sémantiky jsou bezvýznamná. Pokud jim je přidělena sémantika, jedná se o poznatek, z něhož dostaneme nějakou informaci na základě, které je možno se rozhodovat (1, s. 2 - 3).



Obr. 1: Data, poznatky, informace (Upraveno dle 1, s. 3)

2.2 Efekt informačního systému v podniku

Otázkou, jaký efekt bude mít informační systém po integraci do podniku, se zabývají při rozhodování vlastníci podniků, manažeři, ale i informační specialisté. Hlavním důvodem je to, že podnik je pak pomocí IS schopen neustále ovlivňovat hodnotu podniku, konkurenční schopnost, vlastní prodávané výrobky, ale také vztahy se svými zákazníky či dodavateli.

Aby podnik mohl zavést IS ve své činnosti podnikání, měl by mít přehled a představu o přínosech, které můžou nastat zavedením či změnou IS. Činnost IS v podniku můžeme v zásadě shrnout na dva typy úloh a to na tzv. „*dělání správných věcí*“ a „*dělat věci správně*“ (2, s. 178).

2.3 Databáze

Databáze je velké úložiště, kde jsou uchovávána veškerá data pro podnikání organizace a jsou sdíleny současně se všemi uživateli této databáze. Celá tato databáze není ve vlastnictví nějakého oddělení nebo uživatele, ale je sdíleným zdrojem informací celé organizace. Data, která se ukládají do sdílené databáze, mají určitá kritéria. Uživatelé využívající tato data požadují, aby integrována s minimálním počtem duplikací (3, s. 37).

Do databáze se ukládají nejen data celé organizace, ale také data popisující uložená provozní data. Proto můžeme říct, že je databáze definována jako sebe popisující kolekce integrovaných záznamů. Data popisující data neboli metadata, se označují jako systémový katalog nebo slovník dat. Díky charakteru metadat se dostává tzv. nezávislost dat, což znamená, že při jakékoliv aktualizaci stávající struktury databáze nebo přidání nové, nebudou aplikace využívající též databázi nijak ovlivněny, Pokud tedy však přímo nesouvisí z danou změnou. Takovou změnou je myšleno například odstranění sloupce z tabulky, který databázová aplikace využívá. Po takovém zákroku musí být aplikační program upraven. Pokud však přidáváme do databáze novou tabulku nebo do ní pouze přidáváme nový sloupec, nejedná se už o zásah do databázových aplikací (3, s. 37).

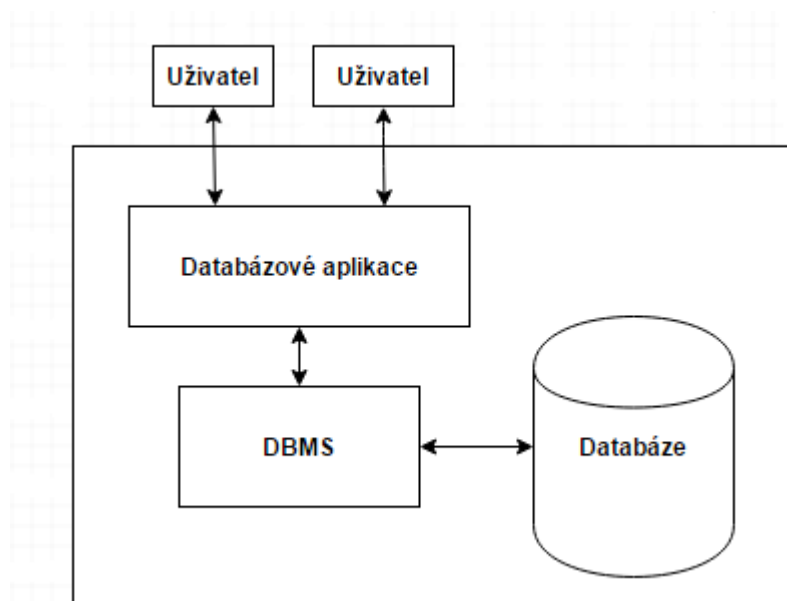
2.3.1 Systém řízení databáze DBMS

Jedná se o software, který nám umožňuje definovat, vytvářet, udržovat databázi, dále také poskytuje řízený přístup k databázi. Další co DBMS uživateli nabízí je možnost vkládat, aktualizovat, mazat a vyvolávat požadovaná data z databáze. Za předpokladu, že existuje centrální úložiště všech a metadat, pak nám DBMS umožňuje dotazování na tyto data pomocí dotazovacího jazyka (3, s. 39).

2.3.2 Databázová aplikace

Databázová aplikace je počítačový program, pomocí kterého jsou uživatelé schopni vytvářet a spravovat databázi a také komunikovat s ní. Takový program může být obvyklou dávkovou aplikací nebo online aplikací, což je častěji se vyskytující možnost v poslední době (3, s. 39).

Obrázek č. 1 nám popisuje, jak jednotliví uživatelé využívají databázové aplikace ke komunikaci s databází pomocí DBMS. Uživatelská aplikace zpracovává datové položky, údržbu dat a generování dat, v DBMS je spravována fyzická struktura a uložení databáze (3, s. 39).



Obr. 2: Komunikace uživatelů s databází (Upraveno dle 3, s. 40)

2.3.3 Architektury DBMS

Architektury DBMS dělíme do několika skupin. Můžeme je rozlišit podle toho, kde probíhá samotná logika provozu, to obstarává program klienta, Dále pak program serveru, kde se spravuje a řídí přístup k databázi. Toto spojení je označováno jako dvouvrstvá architektura (nebo také klient/server), kterou můžeme rozdělit jako architekturu soustředěnou na serveru nebo soustředěnou u klienta (3, s. 41 - 43).

Architektura soustředěná na serveru znamená, že u klienta běží pouze uživatelské rozhraní neboli uživatelská aplikace a logika provozu, zpracování dat a datové služby jsou soustředěny na serveru. Zatímco architektura soustředěná u klienta odpovídá tomu, že na serveru se nachází pouze databáze a datové služby, vše ostatní se zpracovává u klienta. Tato varianta požaduje vyšší nároky na hardware ze strany klienta pro spolehlivý chod (4, s. 8).

Dále je tu třívrstvá architektura, jak už je možno poznat z názvu, je architektura rozdělena mezi tři vrstvy, z nichž každá běží na jiné platformě. První vrstva uživatelského rozhraní se nachází na počítači koncového uživatele. Dále je zde vrstva logiky provozu a zpracování dat, která se zpracovává na serveru, často označovaném jako aplikační server. Na databázovém serveru se nachází už jen databáze a databázové služby. Na tomto serveru může běžet i DBMS, který uchovává data pro střední vrstvu (3, s. 41 - 43).

2.4 Databázové modely

Z hlediska způsobu ukládání dat a vazeb mezi nimi dělíme databáze do následujících modelů.

2.4.1 Lineární model

Jedná se takový model, který je možno implementovat na jakémkoliv systému. Důvodem je to, že tabulky obsažené v tomto modelu mezi sebou nemají žádnou vazbu, tudíž není možné zjistit při situaci dvou tabulek zákazník a zboží, které zboží si zákazník koupil.

Každý záznam zde tedy je tzv. věta databázového souboru. A můžeme říct, že mezi těmito větami neexistuje žádný vzájemný vztah až na ty, co jim následují a předcházejí (4, s. 21).

2.4.2 Relační model

Tento model k nejpoužívanějším datovým modelům současnosti. Za pomoci složení několika lineárních modelů nám vznikne relační model, avšak musí být propojen pomocí položky, nazývané relační klíč. Jedná se ale o krátkodobé spojení. Jednotlivé tabulky vzájemnou relaci ukončí, jakmile přestaneme pracovat s modelem, ale opět vzniká, potřebujeme-li mít propojená data k dispozici (4, s. 21).

2.4.3 Objektový model

Jedná se o nejnovější datový model, ve kterém jsou datové modely postaveny na objektu, který může být brán jako věta, ale jsou k němu přidefinovány metody založené na chování objektu. V objektovém modelu mimo jiné se mohou vyskytovat i relační vazby (4, s. 22).

2.5 Relační datový model

Tento model jsem si vybral jako výchozí pro moji práci, jelikož svými vlastnostmi pokrývá funkčnost plánované databáze je tedy vhodným kandidátem. Jak již víme, relační model propojuje různé tabulky za pomoci relačních klíčů, jimiž jsme schopni získávat data z více tabulek a jejich vzájemné vztahy. Tato kapitola nám více přiblíží vlastnosti tohoto modelu.

2.5.1 Relace

Databáze ukládají údaje formou relací, jenže je speciálním typem tabulky. Relaci je možno popsat jako dvourozměrnou tabulku, která se vyznačuje řádky a sloupci. Vlastnosti relace jsou shrnuty následovně.

- „1. Řádky obsahují data o entitě,*
- 2. Sloupce obsahují data o attributech entity,*
- 3. Buňky v tabulce uchovávají jedinou hodnotu,*
- 4. Všechny položky ve sloupci jsou stejného druhu,*
- 5. Všechny sloupce mají jedinečný název,*
- 6. Na pořadí sloupců nezáleží,*
- 7. Na pořadí řádku nezáleží,*
- 8. Žádné dva řádky nesmí obsahovat identické sady datových hodnot.“ (5, s. 78)*

2.5.2 Integritní omezení pro entity

Jedná se o pravidla, která zajišťují správnost ukládání dat. Tato pravidla jsou rozdělena mezi následující integrity:

- Doménová integrita – zajišťuje, aby všechny hodnoty daného atributu byla v množině povolených a přípustných hodnot,
- Entitní integrita – definuje, že každá relace má mít určen primární klíč, jenž určuje každý řádek relace,
- Referenční integrita - udává, že cizí klíč používaný danou relací odkazuje na relaci, která má identický primární klíč s klíčem cizím (4, s. 28).

2.5.3 Typy klíčů

Pojmem klíč zde definuje jeden nebo více sloupců relace, pomocí něhož jsme schopni identifikovat řádek. Tento klíč může jedinečný tak i nejedinečný. Rozlišujeme tři základní typy klíčů (5, s. 81).

- Kandidátní klíč – jedná se o klíče, které jsou jedinečné a identifikují každý řádek v relaci,
- Primární klíč – je to kandidátní klíč, který je zvolen následně jako primární a systém, s jehož pomocí je schopen identifikovat všechny řádky relace,
- Cizí klíč – je takový klíč, který je atributem jiné relace a uchovává hodnoty (5, s. 86).

2.5.4 Integritní omezení pro vztahy entit

Toto omezení omezuje kardinalitu vztahů mezi následující poměry:

- Vztah 1:1 – tento poměr nám říká, že k jedné n-tici relace odpovídá vždy jedna nebo žádná n-tice relace. Například člověk může vlastnit jeden nebo žádný občanský průkaz,
- Vztah 1:N – zde je poměr naznačuje, že k jedné n-tici relace odpovídá jedna či více n-tic jiné relace,
- Vztah N:1 – vztah N:1 je stejný jako předcházející a je možno ho popsat jako, že 1 nebo více n-tic relace odpovídá pouze jedné n-tici jiné relace,
- Vztah N:M – Poslední poměr je definován tak, že několika n-ticím jedné relace odpovídá jedna nebo více n-tic jiné relace (4, s. 32).

2.6 Návrh databáze

Metodologie návrhu databáze popisuje jednotlivé kroky k vytvoření databázového systému. Tento návrh se skládá ze tří fází: konceptuální návrh, logický návrh a fyzický návrh. Každá z těchto fází obsahuje dílčí úkoly, které je nutno dodržet.

2.6.1 Konceptuální návrh

Tato fáze má za úkol vytvořit ER model, který vychází z datových požadavků organizace nebo její části. V tomto modelu by se měly objevit entity a relace mezi nimi. Dále musí obsahovat atributy, jejich domény, primární a alternativní klíče a nakonec integritní omezení (3, s. 207).

Aby ER model mohl být zhotoven, je nutné vykonat následující úkoly:

„1.1 identifikace entit,

1.2 identifikace relací,

1.3 identifikace a spojení atributů s entitami,

1.4 určení domén atributů,

1.5 určení atributů, které budou kandidátními, primárními a alternativními klíči,

1.6 specializace/generalizace entit (volitelný krok),

1.7 kontrola redundance v modelu,

1.8 kontrola, zda model podporuje uživatelské transakce,

posouzení konceptuálního návrhu databáze s uživateli.“ (3, s. 208)

2.6.2 Logický návrh

V logickém návrhu dochází k tvorbě relačních tabulek podle ER modelu vytvořeném v předcházející části. Každá tabulka by měla být zkontrolována, zda je náležitě strukturovaná. Kontrola se provádí pomocí normalizace, ta zabrání, že nedojde k duplicitě údajů. Díky to se zajistí, že tabulky jsou schopné podporovat uživatelské transakce. Mimo kontroly struktury tabulek se nakonec provádí formulace integritních omezení pro zajištění všech omezení v logickém návrhu (3, s. 207).

Logický návrh se sestává z těchto kroků:

„2.1 vytvoření tabulek,

2.2 kontrola tabulek pomocí normalizace,

2.3 kontrola, zda tabulky podporují uživatelské transakce,

2.4 kontrola integritních omezení,

2.5 posouzení logického návrhu databáze s uživateli.“ (3, s. 208)

2.6.3 Fyzický návrh

U této fáze je potřeba se rozhodnout, jak správně převést logický návrh na fyzický návrh databáze, to znamená tabulky, sloupce a integritní omezení. Tento návrh je pak možno implementovat na cílový DBMS, a protože je velká část fyzického návrhu závislá na tomto DBMS, potřeba dobře znát jeho funkčnost (3, s. 207).

Třetí fáze návrhu:

„3.1 návrh podkladových tabulek,

3.2 návrh reprezentace odvozených dat,

3.3 návrh zbývajících integritních omezení.“ (3, s. 208)

Fáze fyzického návrhu obsahuje ještě dalších pět kroků:

„krok 4 volba organizace souborů a indexů,

krok 5 návrh uživatelských pohledů,

krok 6 návrh bezpečnostních mechanismů,

krok 7 zvážení zavedení kontrolované redundance,

krok 8 monitorování a doladění systému v provozu.“ (3, s. 208)

2.7 Normalizace

Normalizace je proces, pomocí něhož jsme schopni eliminovat duplicitní a nekonzistentní údaje v relačních databázích, které by mohly vést ke ztrátě integrity dat. Data jsou normována použitím několika normálních forem a platí, že čím vyšší forma, tím více jsou data strukturována (6).

Normalizace obsahuje celkem šest úrovní normalizace neboli normální formy (NF). První tři úrovně sledují funkční závislosti mezi atributy relace. Pak je tu Boyce-Coddova forma, ta silněji definuje 3NF. A nakonec 4 a 5NF, ty jsou založené na vícehodnotových závislostech. Ve většině případů stačí provádět normalizaci pouze do 3NF, z důvodu zachycení většiny problémů již prvních normalizačních úrovních (7, s. 89).

Pokud tabulka nepodléhá ani 1NF, pak jde o nultou normální formu, ta z hlediska dotazování představuje téměř neřešitelné rébusy (7, s. 89).

2.7.1 První normální forma

Pokud tabulka obsahuje pouze atomické nebo dále nedělitelné atributy, pak splňuje podmínku první normální formy. Příklad takového rozdělení je, že máme jméno Josef Dvořák, v tomto případě se jedná o složený atribut a je nutno jej rozdělit na jméno a příjmení. Pak už je splněna 1NF (7, s. 90).

2.7.2 Druhá normální forma

Každý atribut musí být zcela závislý na celém primárním klíči. Pokud je tato podmínka splněna a zároveň je splněna podmínka 1NF, jedná se tedy o 2NF. Tato forma se týká pouze tabulek, které mají více primárních klíčů. To znamená, že pokud obsahuje tabulka jeden primární klíč, pak automaticky splňuje podmínku 2NF (7, s. 92).

2.7.3 Třetí normální forma

Tabulka se nachází ve třetí normální formě, splňuje-li opět předchozí normalizační úroveň a také pokud jsou odstraněny všechny neklíčové atributy, které nejsou závislé na primárním klíči (7, s. 92-93).

2.8 Jazyk SQL

Jazyk SQL je standartní dotazovací jazyk, jehož pomocí jsme schopni komunikovat s relačními databázemi. Z hlediska možnosti tvorby databázových dotazů se jedná o nejrozšířenější jazyk (5, s. 33).

Tento jazyk není vhodný k obecnému programování aplikací, které jsou například u systémů příjmu objednávek nebo ke zpracování mezd. Spíše je vhodný k administraci relační databáze. Běžně se jazyk užívá v kombinaci s výše uvedenými procedurálními a objektově orientovanými jazyky (8, s. 33).

2.8.1 Datové typy

Jazyk SQL nám umožňuje definovat takové typy jako jsou numerické hodnoty, znakové či bitové řetězce. Ale také temporální a data časové intervaly. V tabulce 1 jsou popsány konkrétní příklady.

Tab. 1: Datové typy SQL (Upraveno dle 9, s. 61)

| | |
|----------------------------|---|
| Přesné numerické typy | INTEGER, SMALLINT, NUMERIC a DECIMAL |
| Aproximační numerické typy | FLOAT, REAL a DOUBLE PRECISION |
| Znakové typy | CHARACTER(n) a CHARACTER VARYING(n) |
| Časové typy | DATE, TIME, TIMESTAMP, TIME WITH TIME ZONE a TIMESTAMP WITH TIME ZONE |

Samozřejmě datových typů je daleko více, datové typy nacházející se v tab. 1 jsou pouze příklady pro objasnění o tomto tématu.

2.9 Microsoft SQL Server 2012

Jedná se databázový a analytický systém vyvinutý společností Microsoft. SQL Server 2012 je zde k dispozici ve třech základních edicích, jejichž vlastnosti jsou rozděleny z hlediska zákaznických potřeb. Jedná se o následující edice, které mají 32bitovou a 64bitovou verzi:

- „*Standard edition*,
- *Enterprise edition*,
- *Business Intelligence edition*.“ (10, s. 24)

Standard edition je základní vydání určené pro firemních oddělení. Obsahuje také prvky Business Intelligence, ale pouze v přiměřeném rozsahu pro menší firmy, neobsahuje tedy funkce, které jsou typické pro velké podniky (10, s. 25).

Enterprise edition je už ucelenější platforma, která splňuje vysoké nároky firemních aplikací, jak pro zpracování transakcí, tak pro datové sklady. Vytváří infrastrukturu s vlastností zpracování velkých množství dat a vysokého podnikového zařízení. Splňuje také otázku bezpečnosti s integrovanými funkcemi na ochranu dat před neoprávněným vniknutím (10, s. 25).

Business Intelligence edition zde nejvíce vystihuje význam Business Intelligence a všeho, čeho se to týká. Tím jsou myšleny analýzy, multidimenzionální databáze a dolovní dat v podnikové praxi (10, s. 26).

2.10 SWOT analýza

Jedná se o typ strategické analýzy stavu firmy či organizace na základě jejich silných a slabých stránek a následných příležitostí či hrozeb, které plynou z podnikové činnosti. Analýzu je možno rozdělit na vnější a vnitřní prostředí. Ve vnitřním prostředí sledujeme současný stav organizace, kdežto ve vnějším se zaměřujeme na současné okolí organizace (11, s. 297).

Ve vnitřním prostředí identifikujeme a hodnotíme silné a slabé stránky podnik, které vymezují vnitřní faktory efektivnosti organizace ve všech důležitých oblastech (11, s. 297).

Ve vnějším prostředí tedy identifikujeme příležitosti a hrozby, které vycházejí z vlivu okolí podniku ve všech podstatných oblastech, ve kterých organizace působí (11, s. 298).

3 ANALÝZA SOUČASNÉHO STAVU

V této kapitole budu především popisovat společnost, pro kterou budu navrhovat databázi. Nejprve se zaměřím na to jaká je úloha a pozice společnosti na trhu. Dále přistoupím k jednotlivým analýzám a na konec nastíním problematiku této společnosti, ze které budu vycházet v návrhové části.

3.1 Představení společnosti

Název: DVOŘÁK TRUCK s.r.o.

Právní forma: Společnost s ručením omezením

Sídlo: Velké Meziříčí, Karlov 1119

Datum zápisu do obchodního rejstříku: 1999

Firma Dvořák Trucks je společnost která se zabývá nákupem, prodejem, dovozem ze zahraničí, servisem a odtahem nákladních automobilů, včetně návěsů a přívěsů osvědčených značek. Vozidla vykupuje od značkových dealerů ze zemí západní Evropy, jedná se především o země Německo, Holandsko a Francie.

Firma vstoupila na trh v roce 1999. Začínalo se nákupem aut ze zahraničí a jejich prodejem jich dál. Prodej se vztahoval pouze na české území, postupem času se začalo prodávat i do zahraničí. Během roku firma začala vozit více aut týdně ze zahraničí a postupně se firma rozrůstala. V roce 2006 se rozšířila o servisní centrum, kde je kapacita 10 míst na stání, mimo to spustila provoz mycí linky.

Roku 2002 společnost otevřela pobočku na Slovensku v Bratislavě, která se zatím zabývá pouze nákupem a prodejem nákladních vozidel, ale plánuje se do budoucna rozšířit.

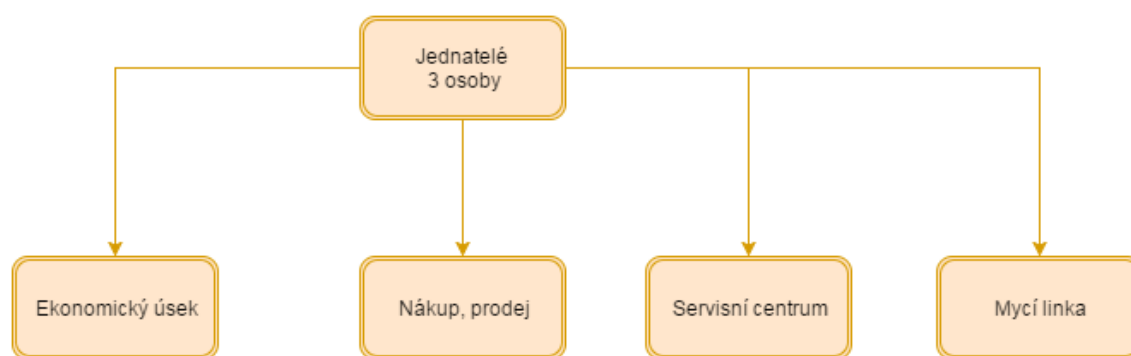
Dále firma poskytuje tyto služby:

- servis,
- pneuservis,
- mycí linky pro osobní a nákladní vozidla,
- ověření, výměny a montáže analogových a digitálních tachografů
- zajištění STK, emise,
- firma také zprostředkovává komisní prodej.

Mezi služby byla zahrnuta i půjčovna vozidel, ale bohužel musela být stažena z důvodů nedbalosti ze strany zákazníku. Vozidla se vracela v poškozeném stavu nebo vůbec.

3.2 Organizační struktura

V tomto případě by se dalo říci, že se jedná o rodinnou firmu a jedná sama za sebe, takže nemá žádné nadřazené subjekty.



Obr. 3: Organizační struktura (Vlastní zpracování)

Ve firmě pracuje zhruba 40 zaměstnanců. Majitel a jednatel společnosti je Daniel Dvořák, dále jsou jednatelem ještě Jaroslav a Stanislava Dvořákové. Každý z nich má na starosti jeden úsek, za který zodpovídá.

3.3 Konkurence

V okolí firmy Dvořák Trucks je velké množství firem, ať už jde jen o prodej vozidel nebo služby, které jsou stejné jako služby firmy Dvořák Trucks.

Vzhledem k veliké konkurenci v okolí se firma snaží vždy vyjít svým zákazníkům vstříc. Jelikož neinvestuje do propagačních reklam, dává firma na staré dobré slovo od svých zákazníků, kteří přišli s firmou do kontaktu. Oproti některým firmám Dvořák Trucks disponuje tím, že sídlí u dálničního sjezdu, tudíž je podstatě nablízku všem zákazníkům, kteří projíždí kolem a potřebují využít služby této společnosti.

3.4 Informační technologie

Než přejdeme k návrhové části, je nutné poznat jaké prostředky a zařízení firma vlastní a používá ke své činnosti podnikání.

3.4.1 Hardware

Ve společnosti se nachází několik kancelářských oddělení, ve kterých jsou umístěny stolní počítače, ale několik zaměstnanců vlastní také notebooky. Co se týče operačního systému, jedná se převážně o Microsoft Windows typu XP Profesional, Windows 7 a na pár výjimek obsahují některá PC operační systém Windows 10. V poslední řadě je zde datové úložiště, které je připojeno k firemní síti. Na tomto úložišti se zálohují firemní dokumenty organizace.

Jak už bylo zmíněno, většina PC se nachází v kancelářích, kde jde o kancelářské činnosti jako například o administrativní či evidenční činnosti, tudíž nejsou na tyto prostředky kladen příliš velké nároky. Další jsou umístěny v servisním centru, kde je potřeba řídit evidenci zákazníků, oprav, sklad zásob atd. Pak PC, které je určeno k pouze k měření tachografů.

Další hardware, který zde stojí za zmínění, se nachází v servisním centru. Jedná se spíš elektronická zařízení sloužící k diagnostice atd. A nakonec je možno sem zařadit samotnou mycí linku.

3.4.2 Software

V tomto ohledu firma využívá několik systémů jako je program VDO, jenž je používán v servisním centru pro měření tachografů a zaznamenávání těchto hodnot. Je zde ještě zastaralý program využívaný pouze centrem, tento systém se však bude nahrazovat novým. Dále jsou to systémy na měření emisí nebo na diagnostiku elektroniky vozidel. V rámci kancelářského softwaru vlastní firma licence pro sadu Microsoft Office a licence antiviru Avast, tyto licence jsou zajištěny pro všechny PC.

3.5 SWOT analýza

Na základě předchozích analýz a podkladů firmy je možno shrnout důležité poznatky ve SWOT analýze, díky tomu jsme schopni zjistit silné a slabé stránky. A následně můžeme identifikovat případné příležitosti, kterých může firma využít, ale také hrozby, kterým musí čelit nebo mohou firmu ohrozit. Tyto poznatky jsou pro lepší přehlednost sepsány do tabulky na další straně.

| Silné stránky (strengths) | Slabé stránky (Weaknesses) |
|--|--|
| <ul style="list-style-type: none"> - Profesionální zaměstnanci a servisní technici | <ul style="list-style-type: none"> - Nedostatečný informační systém - Evidence smluv v papírové podobě - Neefektivní tok informací ve firmě - Náročná evidence zásob |
| Příležitosti (Opportunities) | Hrozby (Threats) |
| <ul style="list-style-type: none"> - Rozšíření nového softwaru - Rozšíření dceřiné pobočky | <ul style="list-style-type: none"> - Jednoduchá ztráta a zneužití dat |

3.5.1 Silné stránky

Další silnou stránkou jsou samotní zaměstnanci, kteří díky dlouholeté praxi ve firmě načerpali tolik zkušeností, díky kterým dokáží vyhovět zákazníkům na vysoké úrovni. Totéž se týká i techniků v servisu, kteří jsou ve svém oboru profesionálové, tudíž své zakázky dokončují kvalitně a včas. To platí i ve věci evidence zaměstnanci jsou zde pečliví a ví přesně, co dělají, ale i zde může lidský faktor selhat.

3.5.2 Slabé stránky

Cíle této bakalářské práce jsou především zaměřeny na slabé stránky firmy. Jedná se hlavně o to, že ve firmě chybí systém, jenž by administrativní práci ulehčil nebo zohlednil. Pokud si data zaměstnanci sdílejí a postupně s nimi manipulují, tak dochází k nekonzistenci, což je nepřipustné pro celou organizaci, proto zde třeba integrace centralizovaného systému.

3.5.3 Příležitosti

Pro tuto firmu je hlavní příležitost získat systém, jenž by ulehčil každodenní procesy celé firmy od administrativních činností až po servisní záležitosti v rámci servisu.

Další příležitost se naskýtá v možnosti rozšíření pobočky v Bratislavě, která postrádá servis, to znamená, že po rozšíření je možno oslovit zcela nové zákazníky.

3.5.4 Hrozby

Co se hrozeb týče, tak asi nejvíce firmu Dvořák Trucks z hlediska evidence ohrožují možnosti ztráty či zneužití dat. Vzhledem k tomu, že je zde nulové zabezpečení, je tento faktor velice pravděpodobný.

3.6 Současný proces zpracování dat

Zde budou popsány procesy, na které se bude databázový model zaměřovat.

3.6.1 Evidence smluv

Při prodeji každého vozidla je vypsána kupní smlouva, která se později eviduje, jak jsem již zmínil v programu Microsoft Excel, což z pohledu administrace přináší několik úskalí z hlediska práce na této smlouvě v několika odděleních. Smlouva jako taková je za archivuje, ale zapisují se důležité údaje o této smlouvě jako je číslo smlouvy, předmět smlouvy atd. Tato smlouva pak později slouží k potvrzení, zde je ještě platná záruka a jiné náležitosti, což pak někdy, kvůli zdlouhavému vyhledávání je dost nepříjemné. Na základě této kupní smlouvy je vystavena faktura o prodeji a oba tyto dokumenty se po podepsání podávají ekonomickému úseku pro další zpracování, kde se údaje o nich zaznamenávají v programu Excel. V rámci těchto údajů pak zaměstnanci z ekonomického úseku kontrolují platební schopnost zákazníků.

Dále je zde smlouva, která je zákazníkům nabízena při každé návštěvě či při prodeji nějakého vozidla. Jedná se v podstatě o členskou neboli klubovou smlouvu. Například po podepsání kupní smlouvy a za symbolický poplatek bude-li kupující souhlasit, tak získává 10% a 5% slevu na vybrané služby v servisu. Toto je možné uplatnit každé dva roky. A zde se nachází problém, když zákazník předá objednávku do servisu, tak se občas stane, že slevu má po splatnosti, samozřejmě jde tuto slevu opět prodloužit, ale zaměstnanci na servisu nemají oprávnění toto vyřizovat, pouze vypíší žádost na ekonomický úsek. Vzhledem k tomu, že jde o dvě komunikace mezi dvěma budovami, tak jakákoliv aktualizace je komplikovaná občas se stane, že tento poplatek, který už je zaplacen, tak je stále vyžadován k úhradě. Navíc aby tato smlouva byla opět platná, tak je nutné ji najít vzhledem k tomu, že evidence opět probíhá v Excelu jen nutné ji dohledávat smlouvu po smlouvě a to vede k častému zdržení. Zákazník také má možnost od té smlouvy odstoupit úplně, tak je možno ji zrušit, aby tomu tak bylo, je zase nutné tuto smlouvu dohledávat a informovat o tom i ostatní pověřená oddělení.

3.6.2 Příjem vozidel na servis a mycí linku

Na základě každé návštěvy servisu vypisuje zaměstnanec se zákazníkem objednávku či zakázku opravy nebo jiné nabízené služby. Tuto objednávku vždy sepisuje vedoucí pracovník v servisu. Do tohoto formuláře se vypisují údaje o zákazníkovi jako je jméno, bydliště atd. Dále pak o jaké služby se bude jednat, případně závady na vozidle. Následně se zadávají technické specifikace vozidla jako je značka a typ vozu, mimo jiné sem zadává i VIN číslo, není to však podmínkou. A nakonec se zde vypíše přibližné ukončení opravy či služby.

Co se mycí linky týče, postup je v zásadě stejný. Na základě dodacího listu se vypíší opět údaje o zákazníkovi a u vozidla se zajímáme pouze o SPZ a o jaký typ se jedná, protože každá souprava vozidla je jiná, tudíž jsou pro ni i rozdílné ceny. Na dodacím listu se tedy vyplní, o jaký typ soupravy jde a následně i program mytí, jenž je pro každý typ soupravy také jinak cenově ohodnocen.

Po řádném vyplnění všech náležitostí předá vedoucí pracovník tyto dokumenty pracovníkům, kteří podle těchto formulářů vědí, jak postupovat a na jakém vozidle. Pro

lepší objasnění, jak tyto formuláře vypadají, tak jsou znázorněny na následujících stranách.

ZAKÁZKA/OBJEDNÁVKA

ČÍSLO: 17210053

| | |
|--|---|
| Dodavatel: DVOŘÁK TRUCK - SERVIS s.r.o Karlov 1119, 594 01 Velké Meziříčí  IČ: 02461587 DIČ: CZ02461587 TEL: +420 566 521 555 MOBIL: +420 604 897 679 | Odběratel: IČ: DIČ: TEL: |
|--|---|

| | |
|-------------------------------|---|
| <i>SERVIS - DVOŘÁK TRUCKS</i> | Karlov 1119, Velké Meziříčí, D1 EXIT 146 tel: 731 185 717, 604 897 679 |
|-------------------------------|---|

| |
|-------------------------|
| Značka: |
| Typ vozidla |
| VIN číslo: |
| Stav km: |
| Důvod objednávky: |
| |
| |

| | | |
|---------------|---------------|---------|
| Datum: | | Razítko |
| Předal: | Předal: | |
| | | |










Obr. 4: Příjmová objednávka na servis (Převzato od Dvořák Trucks)

DODACÍ LIST

číslo: 0007001^R

| | |
|---|-------------------------------|
| Dodavatel: DVOŘÁK TRUCK - SERVIS s.r.o. Karlov 1119, 594 01 Velké Meziříčí IČ: 02461587 DIČ: CZ02461587 TEL: +420 566 521 555 Mobil: +420 604 897 679 | Odběratel: IČ: DIČ: |
|---|-------------------------------|

| | |
|-----------------------------------|---|
| MYCÍ LINKA - DVOŘÁK TRUCKS | Karlov 1119, 594 01 Velké Meziříčí, D1 EXIT 146 tel.: 731 185 717, 604 897 679 |
|-----------------------------------|---|

| Ceny jednotlivých mycích programů bez DPH | | | | | | | | | |
|---|---|---|--------------------|---|----------------|---|-----------------------------|---|-------------------------------|
| Mycí programy: | | NORMÁL šampón kartáč oplach, sušič | CHEMIE předmytí | | Spodní mytí | | Boční mytí vysokotlak | | Odstranění hmyzu chemie |
| osobní automobil |  | 90,- | 30,- | 6 | 25,- | 2 | 20,- | 7 | 25,- |
| dodávka do 5 m |  | 130,- | 30,- | 6 | 25,- | 2 | 20,- | 7 | 25,- |
| autobus do 9 m |  | 300,- | 80,- | 1 | 60,- | 2 | 50,- | 3 | 40,- |
| tažná souprava |  | 480,- | 120,- | 1 | 60,- | 2 | 50,- | 3 | 40,- |
| valník + přívěs |  | 500,- | 120,- | 1 | 60,- | 2 | 50,- | 3 | 40,- |
| kabina řidiče |  | 270,- | 40,- | 6 | 20,- | 2 | 20,- | 7 | 40,- |
| valník |  | 300,- | 60,- | 1 | 30,- | 2 | 30,- | 7 | 40,- |
| speciální návěsy (cisterna ...) |  | 590,- | 80,- | 1 | 60,- | 2 | 50,- | 3 | 30,- |
| autobus se zrcátky |  | 380,- | 80,- | 1 | 60,- | 2 | 50,- | 3 | 40,- |
| - příplatek za ruční domytí nestandardních návěsů (cisterna ...) 250,- Kč | | | | | | | | | |
| - v ceně je zahrnutý oplach rámu a umytí disků | | | | | | | | | |

| | | | |
|---------------|--------------------------------|------------|------------------|
| Cena bez DPH | DPH Kč | Cena s DPH | Razítko, podpis: |
| SPZ: | Převzal: | | |
| Mytí provedl: | Datum uskuteč. zdaň plnění: | | |

Obr. 5: Příjmoví dodací list k mycí lince (Převzato od Dvořák Trucks)

3.6.3 Předání vozidla zpět zákazníkovi

Po skončení oprav, případně vyčistění daného vozidla kontaktuje firma majitele zákazníka, který vozidlo předal. V tuto chvíli, pokud zákazník není smluvní tak je mu nabídnuta smlouva pro využívání slev do budoucna. Tento postup je popsán v článku 3.6.1. Zákazníkovi se vystaví seznam případných závad a činností vykonaných na vozidlo a spolu s tím je mu vystavena faktura, ve které je už započítaná práce zaměstnanců, použitého materiálu či možných slev atd. Tato faktura je oproti prodejní faktuře vypisovaná ručně přímo na servisu.

3.7 Shrnutí situace ve firmě

Jak již bylo zmíněno, firma nedisponuje kvalitním informačním systémem. Z toho vyplývá hned několik problému hlediska evidence. Bohužel firma tedy využívá papírovou evidenci nebo uchovává své údaje formou Excelu, z čehož vyplývá, že administrace je nepřehledná a aktualizace dat je takřka nemožná v rámci celé organizace.

Když se zaměříme na servisní centrum a fungování evidence v něm, tak zjistíme, že se zde využívá zastaralý program, který nesplňuje požadavky organizace po funkční stránce a chybí mu účetní oblast na vystavování faktur. Administrace dat v tomto programu nebyla nijak zvlášť složitá, ale v rámci celé organizace jde o decentralizovaný systém a vyhledávání potřebných údajů je pak složité. Co se týče skladu zásob a náhradních dílů je těžká orientace v zásobách, z důvodu chybějícího modulu pro evidenci zásob. Tím pádem evidence se vede papírovými záznamy.

Vezmeme-li v úvahu kancelářskou činnost, tak zde půjde především o nákup a prodej vozidel. Evidence opět je založena na využívání programu Excel, zde se ukládají podklady ke smlouvám o prodejkách, nákupech, informace o dodavatelích vozidel a informace o zákaznících. Jelikož je občas potřeba dohledat údaje nějaké informace ať už o vozidle nebo o zákazníkovi, tak jde o náročný proces vyhledávání v několika souborech. Dále, pokud nastane aktualizace některých dokumentů, tak není možné v reálném čase aktualizovat ostatní soubory stejného typu, které využívají i ostatní zaměstnanci.

Posledním stěžejním problém je bezpečnost. Do databáze by měli mít přístup jen vedoucí pracovníci a zaměstnanci k tomu pověření s tím, že se zaměstnanec bude moci přihlašovat pouze do oblasti povolené vedením.

Úkolem této práce tedy bude navrhnout a vytvořit databázový model, který bude evidovat všechny nákupní a prodejní zakázky vedené konkrétním zaměstnancem. Na základě smluv o prodeji bude model vystavovat faktury zákazníkům. Fakturace bude také zaměřena na servisní centrum, kde se budou řídit jednotlivé služby zákazníkům. Do modelu bude zahrnuta také evidence náhradních dílů pro lepší administraci.

4 VLASTNÍ NÁVRHY ŘEŠENÍ

V této kapitole je nastíněna stěžejní část celé práce. Nejdříve zde nastíním důležité aspekty, jak by měla celé databáze vypadat, respektive co by měla obsahovat. V dalších podkapitolách jsou uvedeny procesy týkající se funkčnosti databáze znázorněny za pomoci vývojových diagramů. Po představní těchto procesů následuje konceptuální a logický návrh, ve kterém jsou popsány veškeré entity a v neposlední řadě je zde fyzická úroveň. Na konec bezpečnostní opatření a zhodnotím přínosy tohoto návrhu firmě.

4.1 Požadavky na databázi

Z předchozích analýz vyplývá, že databáze by měla pokrýt veškerou funkcionalitu podniku. Napřed by měla vést záznamy o osobách a to jak o zaměstnancích pracujících ve firmě, zákaznících, kteří přišli s firmou do styku, ale také i o dodavatelích, kteří doplňují firmě sklady ať už vozidly či náhradními díly. Asi nejdůležitější částí celé databáze bude oblast prodeje, kde bude firma moci vidět, jaká vozidla jsou k prodeji, a která už byla prodána. Na to budou navazovat údaje o kupních smlouvách, které se uskutečňují v rámci prodeje vozidla.

Jelikož se jedná o firmu, která nabízí služby v rámci servisu a mycí linky, tak je zde zahrnuta i tato oblast, díky níž bude jasné kdy a pro koho byly dané služby uskutečněny. To znamená, že se budou záznamy typu, jaké práce proběhly na vozidle náležitěmu danému zákazníkovi, který zaměstnanec práci vykonal.

Poslední oblastí budou faktury, jenž se budou vystavovat zákazníkům na základě kupních smluv a objednávek ze servisu a mycí linky a následně budou tyto doklady zaevidovány v databázi.

Nakonec by měla databáze podléhat otázce bezpečnosti, což znamená, aby se do databáze nedostali neoprávnění uživatelé. A také aby zaměstnanci, kteří mohou užívat databázi, měli přístup pouze do sekcí, které spadají k jejich pozici.

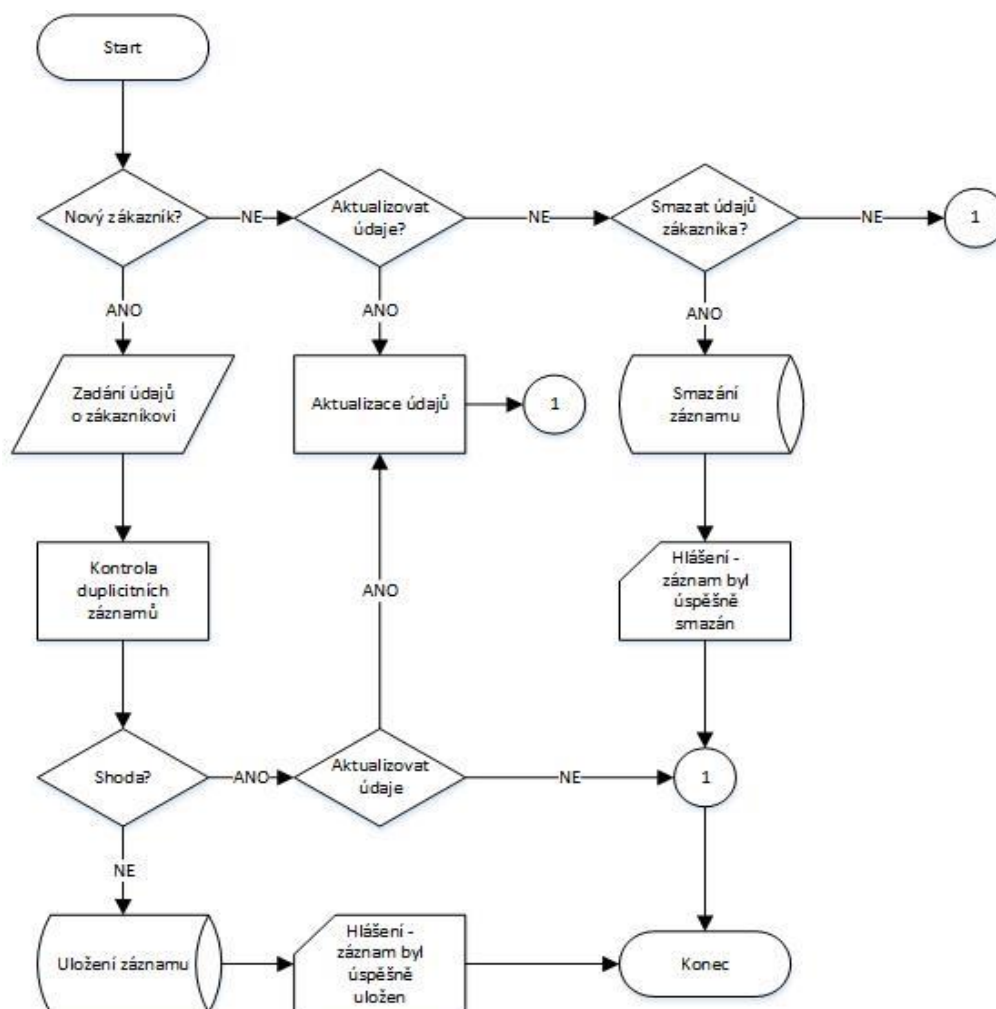
4.2 Interní procesy

Tato kapitola zahrnuje základní procesy ze, kterých se bude vycházet v návrhové části databáze při tvorbě konceptuálního schématu. Jedná se o takové procesy, které náleží dosavadní situaci ve firmě. Samozřejmě zde budou zařazeny jen základní procesy.

Tyto procesy jsou znázorněny pomocí vývojových diagramů. K tvorbě těchto vývojových diagramů a stejně tak i diagramů konceptuální a logické úrovně jsem použil program Visio Profesional 2013.

4.2.1 Vložení nového zákazníka

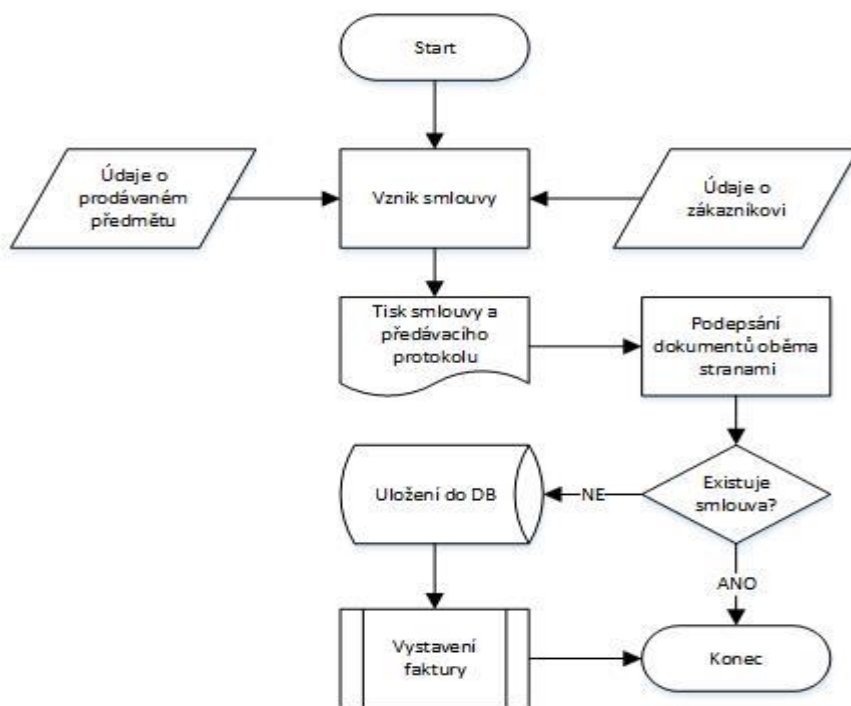
Na následujícím obrázku je znázorněn vývojový diagram, který slouží pro správu všech zákazníků, se kterými firma byla v kontaktu. Na začátku má zaměstnanec na výběr, zda chce vložit novou osobu či ne. To je pak doprovázeno kontrolou, pokud by se v databázi vyskytoval stejný záznam. Nebo také je zde možnost aktualizace, že by se u zákazníka vyskytla nějaká změna. Vymazání zákazníka se stane pouze za situace chybných údajů, jinak záznamy mohou sloužit do budoucna. Co se týče správy dodavatelů a zaměstnanců tak ty zde už nezobrazuji z toho důvodu, že by byly příliš stejné.



Obr. 6: Vývojový diagram procesu evidence zákazníka (Vlastní zpracování)

4.2.2 Vytvoření kupní smlouvy

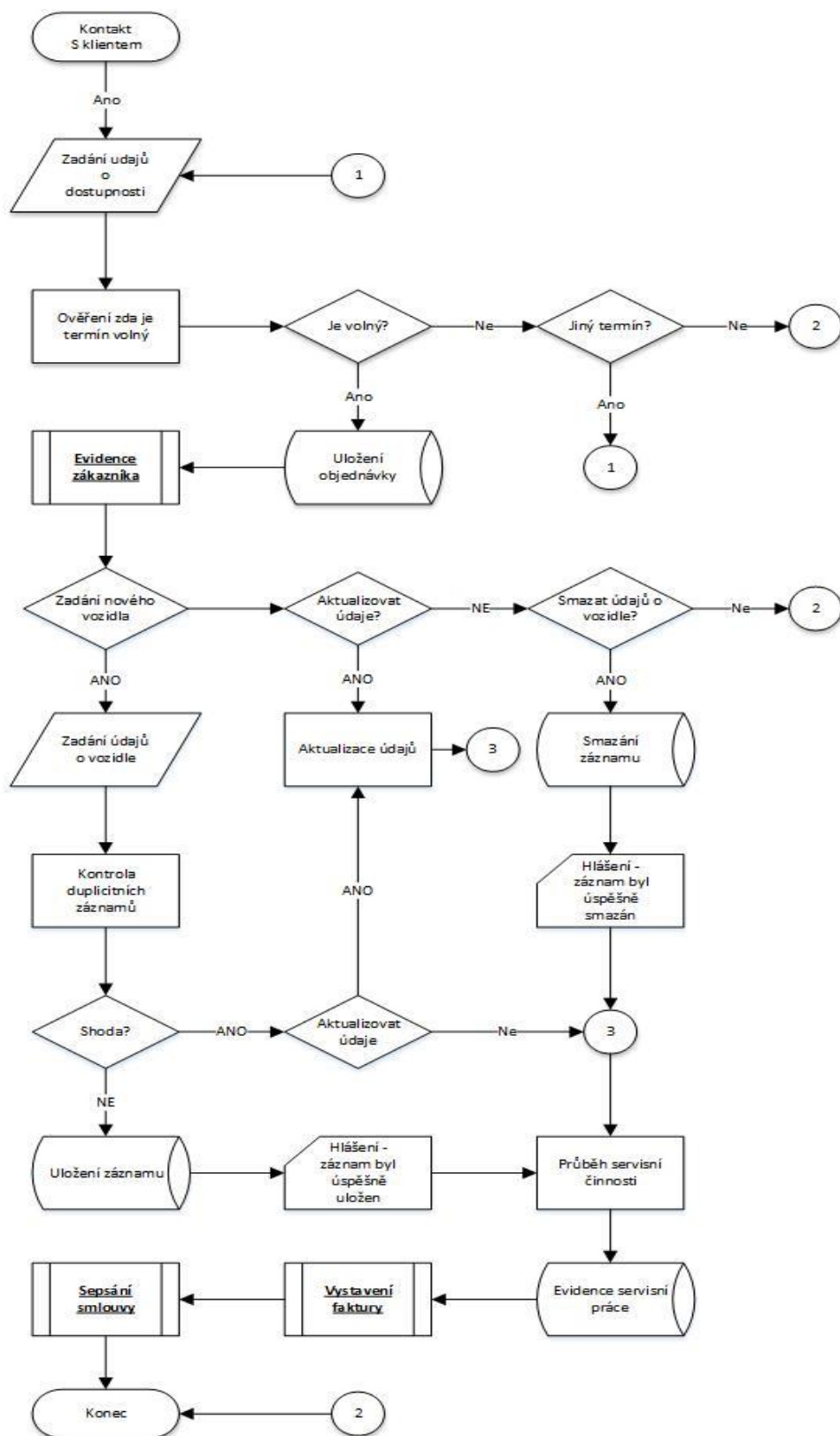
Na obr. 7 je možno vidět vznik kupní smlouvy spolu s předávacím protokolem. Tyto dokumenty slouží pro případ peněžních či jiných nesrovnalostí se zákazníkem. Zrušení smlouvy je zde zcela zbytečné, protože nejde o smluvní vztah, ale pouze o podmínky obchodu a předávací protokol slouží k potvrzení převzetí vozidla zákazníkem. Po sepsání a zaevidování dokumentu je vystavena faktura v podprocesu vystavení faktury, který je popsán pomocí obr. 9.



Obr. 7: Vývojový diagram pro tvorbu kupní smlouvy (Vlastní zpracování)

4.2.3 Příjem vozidla na servis a mycí linku

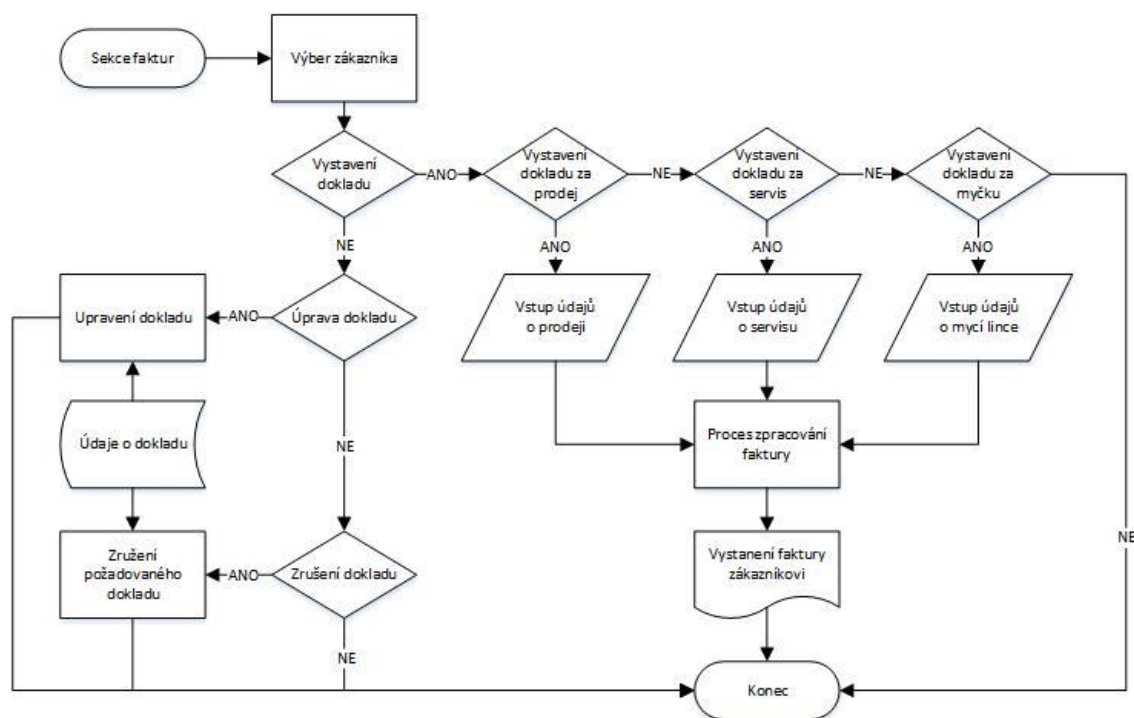
Zde se řeší příjem vozidel. Nejprve se rozhoduje volnost termínů servisní služby, a zda termín zákazníkovi vyhovuje, pak následuje podproces evidence zákazníka, viz obr. 6. Dále je zde zaevidování vozidla ať už jde jen o vložení nového či aktualizace stávajícího. Po vykonání servisní činnosti se veškeré údaje o této záležitosti zaevidují. Na základě toho je pak možno vystavit fakturu, což je zde znázorněno pomocí podprocesu vystavení faktury. Tento diagram je řešen v rámci servisu, ale pro mycí linku by byl stejný.



Obr. 8: Vývojový diagram procesu pro příjem k servisu (Vlastní zpracování)

4.2.4 Sekce faktur

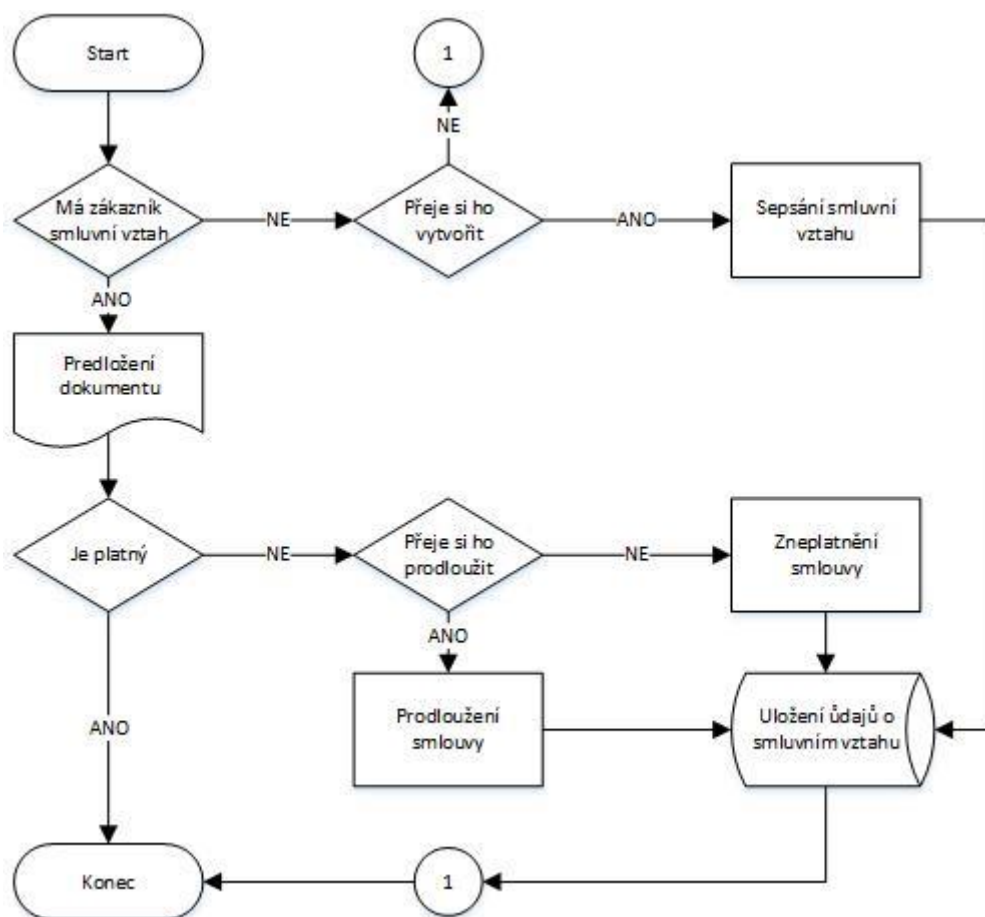
Asi stěžejní částí celé funkcionality bude vystavování faktur, kde je možno vystavit doklady ve všech požadovaných oblastech firmy, kterými jsou prodej, servis a mycí linka. Zaměstnanec zde má mimo vystavení dokladu také možnost upravit či úplně zrušit. Tento doklad se zákazníkovi předává vždy v tištěné podobě.



Obr. 9: Vývojový diagram sekce faktur (Vlastní zpracování)

4.2.5 Vytvoření smluvního vztahu

Poslední vývojový diagram je zobrazen na obr. 10 a vyjadřuje vytvoření smluvního vztahu mezi firmou a zákazníkem. Má-li zákazník zájem, vytvoří se tento dokument. Pokud už zákazník má smluvní vztah, kontroluje se jeho platnost, případně se prodlužuje pro další zákaznické výhody.

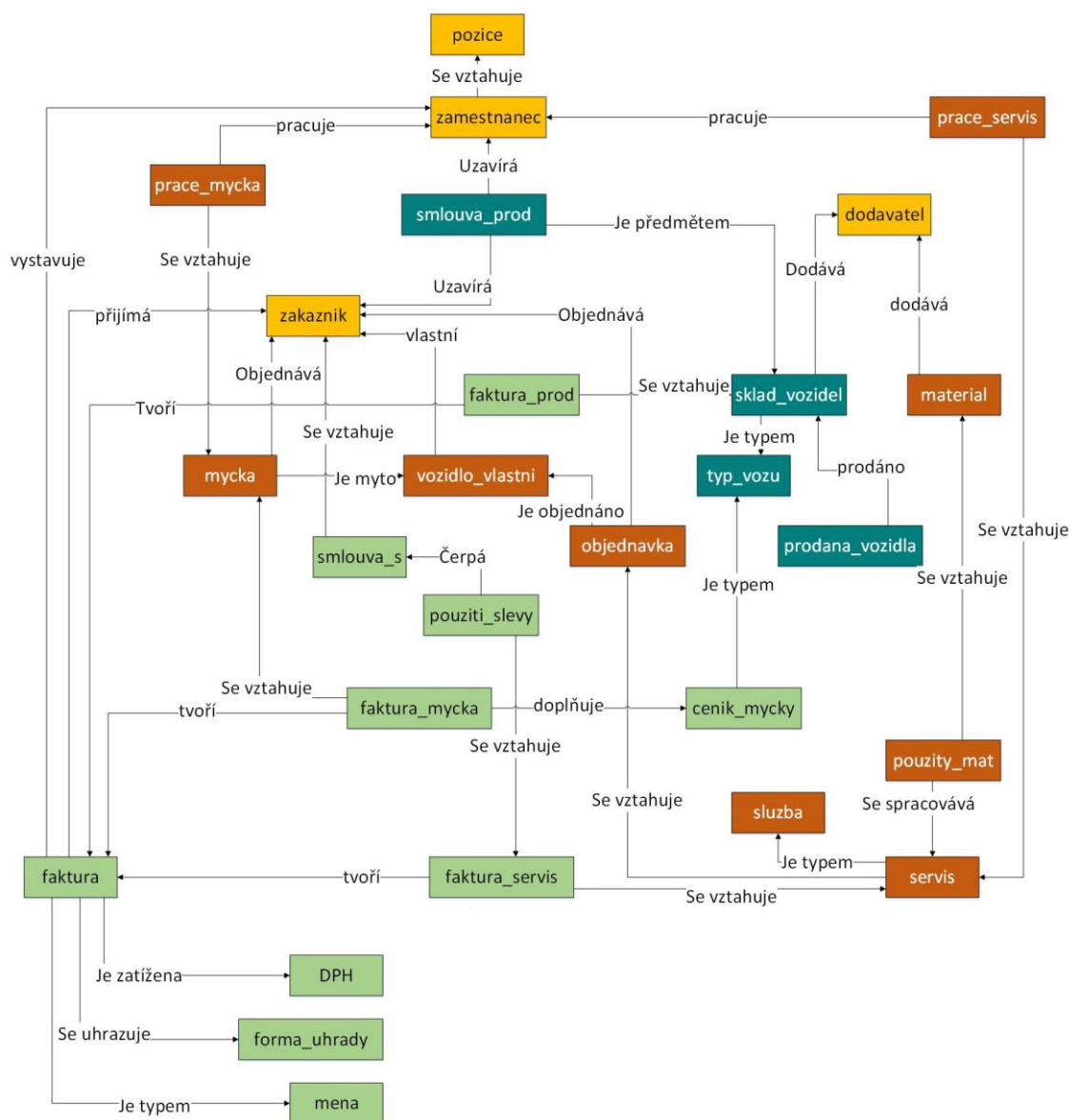


Obr. 10: Vývojový diagram pro vytvoření smluvního vztahu (Vlastní zpracování)

4.3 Konceptuální návrh databáze

Cílem této je zapotřebí identifikovat veškeré entity, které jsou v databázi zapotřebí, dále je důležité zjistit relace mezi těmito entitami. Na následujícím obr. jsou tyto prvky znázorněny pomocí entito-relačního diagramu. Jednotlivé entity jsou zde zobrazeny pomocí obdélníků a relace mezi nimi jsou popsány tak, aby byl jasný vztah mezi entitami. Pro lepší přehlednost zde ještě nejsou určeny jak primární klíče, tak cizí klíče, ale také atributy těchto entit.

Vzhledem k velkému množství entit, je model rozdělen do několika sekcí, které jsou barevně odděleny. Jedná se o sekci osob, faktur, prodeje, servisu a mycí linky.

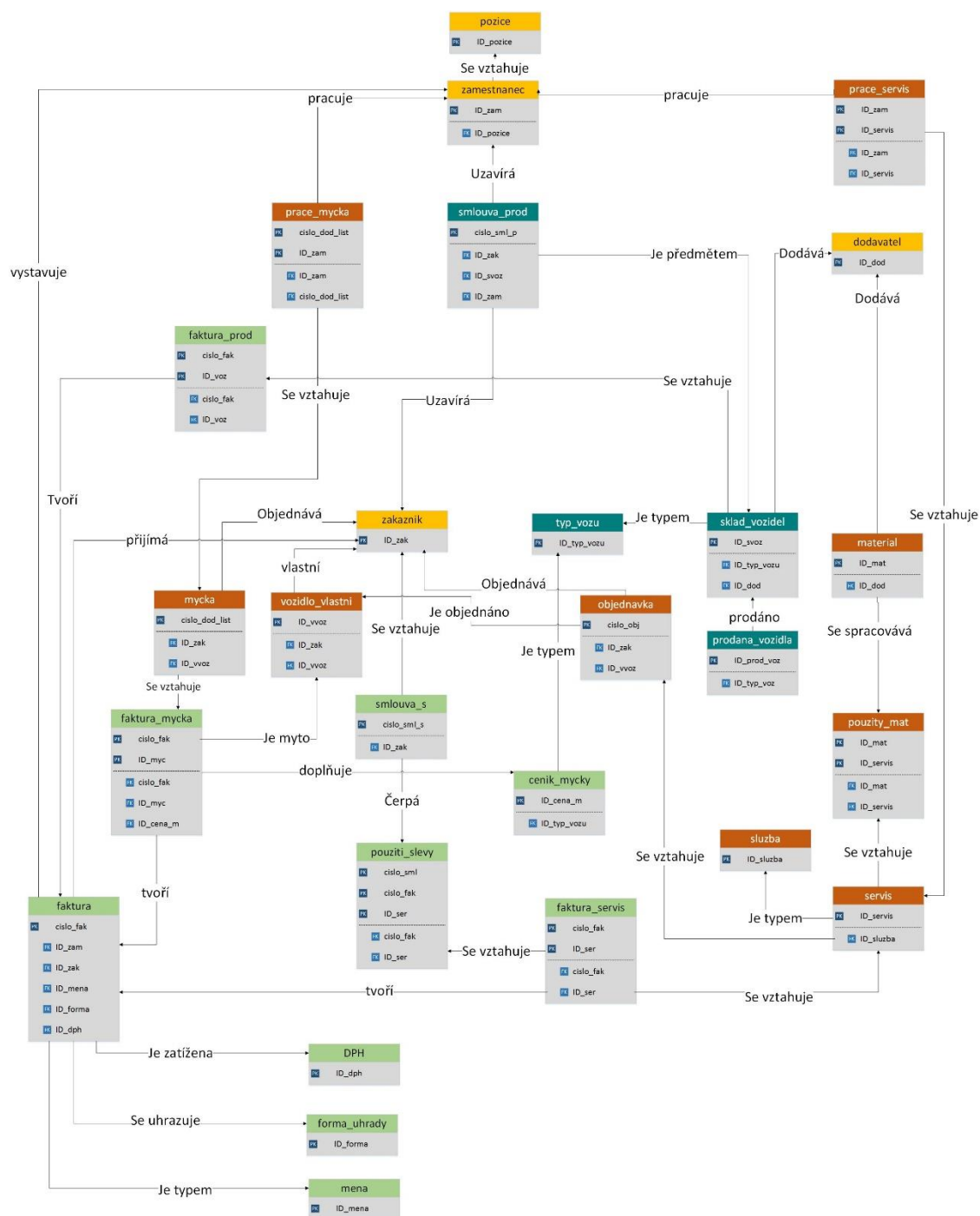


Obr. 11: ER-diagram konceptuální úrovně (Vlastní zpracování)

4.4 Logický návrh databáze

Na základě konceptuálního návrhu je možné přejít k další části a to k logickému návrhu. Díky ER diagramu je možno jednotlivé entity přenést na relační tabulky vytvořit z nich diagram logického návrhu. Následující obrázek tedy obsahuje diagram tabulek obsažených v databázi. Aby byl model přehlednější, tak tabulky obsahují pouze primární a cizí klíče z důvodu velkého množství tabulek. Co se týče atributů obsažených

v relačních tabulkách, tak ty jsou podrobně rozepsány spolu s datovými typy v tabulkách nacházejících podkapitolách.



Obr. 12: Diagram logické úrovně (Vlastní zpracování)

4.4.1 Sekce osob

Tabulka *Pozice* je zde brána pouze jako číselník všech pracovních pozic, které se ve firmě vyskytují. Tabulka se váže k tabulce *Zamestnanec*.

Tab. 2: Pozice (Vlastní zpracování)

| Položka\Tabulka | Pozice |
|-----------------|----------------|
| ID_pozice (PK) | Int (not null) |
| nazev | Varchar (20) |
| Popis | Text |

V této tabulce jsou obsaženy potřebné údaje o zaměstnancích, jako je jméno, bydliště atd. Navíc obsahuje přihlašovací údaje do systému. Tabulka *zamestnanec* je pomocí klíče *ID_pozice* provázána s tabulkou *pozice*.

Tab. 3: Zaměstnanec (Vlastní zpracování)

| Položka\Tabulka | Zamestnanec |
|-----------------|----------------|
| ID_zam (PK) | Int (not null) |
| Jmeno | Varchar (30) |
| Prijmeni | Varchar (30) |
| Cislo_OP | Varchar (9) |
| Email | Varchar (60) |
| Telefon | Varchar (13) |
| Mesto | Varchar (20) |
| Ulice | Varchar (20) |
| Cp | Integer |
| PSC | integer |
| Logi | Varchar (20) |
| Heslo | Varchar (20) |
| ID_pozice (FK) | Int |

Tabulky *dodavatel* a *zakaznik* se v ničem neliší. V obou jsou potřebné údaje k identifikaci dané osoby. Tabulka *zakaznik* je rozšířena pouze o atribut *smluv_zak*, pomocí kterého je možno snadno zjistit, zda zákazník může využít slevu či ne.

Tab. 4: Zákazník (Vlastní zpracování)

| Položka\Tabulka | Zakaznik |
|-----------------|----------------|
| ID_zak (PK) | Int (not null) |
| Firma | Varchar (50) |
| Jmeno | Varchar (30) |
| Prijmeni | Varchar (30) |
| Stat | Varchar (20) |
| Telefon | Varchar (12) |
| Mesto | Varchar (20) |
| Ulice | Varchar (20) |
| Cp | Integer |
| PSC | integer |
| IC | Integer |
| DIC | Integer |
| Smluv_zak | bit |

Tab. 5: Dodavatel (Vlastní zpracování)

| Položka\Tabulka | Dodavatel |
|-----------------|----------------|
| ID_dod (PK) | Int (not null) |
| Firma | Varchar (50) |
| Jmeno | Varchar (30) |
| Prijmeni | Varchar (30) |
| Telefon | Varchar (12) |
| Stat | Varchar (20) |
| Mesto | Varchar (20) |
| Ulice | Varchar (20) |
| Cp | Integer |
| PSC | integer |
| IC | Integer |
| DIC | Integer |

4.4.2 Sekce faktur

Tabulky *forma_uhrady*, *mena* a *DPH* zde slouží opět jako číselníky pro více možností úhrady faktur nebo pro výběr měny níž se bude plaitit.

Tab. 6: Forma úhrady (Vlastní zpracování)

| Položka\Tabulka | Forma_uhrady |
|-----------------|----------------|
| ID_forma (PK) | Int (not null) |
| nazev | Varchar (15) |

Tab. 7: Měna (Vlastní zpracování)

| Položka\Tabulka | Mena |
|-----------------|----------------|
| ID_mena (PK) | Int (not null) |
| zkratka | Varchar (4) |
| nazev | Varchar (20) |

Tab. 8: DPH (Vlastní zpracování)

| Položka\Tabulka | DPH |
|-----------------|----------------|
| ID_dph (PK) | Int (not null) |
| hodnota | Integer |

V tabulce *faktura* se uchovávají všechny podstatné údaje o těchto vystavených dokumentech tedy jen ty základní. Ostatní údaje týkající se předmětu vystavené faktury jsou uvedeny jednotlivých předmětových fakturách *faktura_prod*, *faktura_mycka* a *faktura_servis*. Dále je zde uvedeno, fakturu vystavuje a kdo ji přijímá.

Tab. 9: Faktura (Vlastní zpracování)

| Položka\Tabulka | Faktura |
|------------------|----------------|
| Cislo_fak (PK) | Int (not null) |
| Datum_vystaveni | date |
| Datum_splatnosti | Date |
| Var_symbol | Integer |
| ID_zam | Int |
| Cena_bez_DPH | Integer |
| Cena_s_DPH | Integer |
| ID_zak | Int |
| ID_mena | int |
| ID_forma | Int |
| ID_DPH | Int |
| zaplaceno | bit |

Tabulky *faktura_prod*, *faktura_servis* a *faktura_mycka* obsahují složený primární klíč, který propojuje tabulku *faktura* s tabulkami *servis*, *mycka* a *sklad_vozidel*.

Tab. 10: Faktura prodeje (Vlastní zpracování)

| Položka\Tabulka | Faktura_prod |
|-----------------|----------------|
| Cislo_fak (PK) | Int (not null) |
| ID_voz (PK) | int |

| | |
|--------------|--------------|
| Predmet_prod | Varchar (20) |
| Cena_bez_DPH | Integer |

Tab. 11: Faktura servis (Vlastní zpracování)

| Položka\Tabulka | Faktura_servis |
|-----------------|----------------|
| Cislo_fak (PK) | Int (not null) |
| ID_ser (PK) | int |
| Predmet_prod | Varchar (20) |
| Cena_bez_DPH | Integer |

Tab. 12: Faktura myčka (Vlastní zpracování)

| Položka\Tabulka | Faktura_mycka |
|-----------------|----------------|
| Cislo_fak (PK) | Int (not null) |
| ID_myc (PK) | int |
| inPredmet_prod | Varchar (20) |
| Cena_bez_DPH | Integer |

V tabulce *smlouva_s* slouží pro evidenci smluv, díky které jsou zákazníci schopni čerpat slevy v servisu.

Tab. 13: Smlouva smluvního vztahu (Vlastní zpracování)

| Položka\Tabulka | Smlouva_s |
|------------------|----------------|
| Cislo_sml_s (PK) | Int (not null) |
| od | date |
| do | Date |
| ID_zak (FK) | int |
| zaplaceno | bit |

Tabulka *pouziti_slevy* je pouze výsledkem dekompozice a obsahuje složený primární klíč tabulek *servis* a *smlouva_s*.

Tab. 14: Použití slevy (Vlastní zpracování)

| Položka\Tabulka | Pouziti_slevy |
|------------------|----------------|
| Cislo_sml_s (PK) | Int (not null) |
| Cislo_fak (PK) | int |
| ID_ser (PK) | int |

4.4.3 Sekce prodeje

Tabulka *typ_vozu* obsahuje číselník, který slouží k identifikaci všech typů vozidla, které firma prodává či přijímá na servis nebo myčku.

Tab. 15: Typ vozu (Vlastní zpracování)

| Položka\Tabulka | Typ_vozu |
|------------------|----------------|
| ID_typ_vozu (PK) | Int (not null) |
| nazev | varchar(20) |

V tabulce *sklad_vozidel*, jsou evidována veškerá vozidla, která jsou určena k prodeji, zároveň zde jsou technické specifikace vozidla počínaje jejich cenou.

Tab. 16: Sklad vozidel (Vlastní zpracování)

| Položka\Tabulka | Sklad_vozidel |
|------------------|----------------|
| ID_svoz (PK) | Int (not null) |
| Win_cislo | Varchar (20) |
| Znacka | Varchar (20) |
| Rok_vyroby | integer |
| barva | Varchar (10) |
| Stav_km | integer |
| vybava | text |
| vysladneno | bit |
| CZK_bez_DPH | Integer |
| EUR_bez_DPH | integer |
| ID_typ_vozu (FK) | int |

Do tabulky *prodana_vozidla* se přenášejí ta vozidla, která jsou prodána respektive vyskládněna.

Tab. 17: Prodaná vozidla (Vlastní zpracování)

| Položka\Tabulka | Prodana_vozidla |
|------------------|-----------------|
| ID_prod_voz (PK) | Int (not null) |
| Win_cislo | Varchar (20) |
| znacka | Varchar (20) |
| Rok_vyroby | Integer |
| vybava | Text |
| CZK_bez_DPH | Integer |
| EUR_bez_DPH | Integer |
| ID_typ_vozu (FK) | int |

Tabulka *smlouva_prod* obsahuje informace o kupní smlouvě kterou zákazník uzavírá chce-li koupit nějaké vozidlo.

Tab. 18: Smlouva prodeje (Vlastní zpracování)

| Položka\Tabulka | Smlouva_prod |
|------------------|----------------|
| cislo_sml_p (PK) | Int (not null) |
| Datum | date |
| ID_zak (FK) | Int |
| ID_zam (FK) | Int |
| ID_svoz (FK) | Int |

4.4.4 Sekce servisu a mycí linky

Tabulka *sluzba* slouží opět jako číselník služeb, které jsou v servisu nabízeny.

Tab. 19: Služba (Vlastní zpracování)

| Položka\Tabulka | Sluzba |
|-----------------|----------------|
| ID_sluzba (PK) | Int (not null) |
| nazev | Varchar (20) |
| popis | Text |
| Cena_za_hod | Integer |

Tabulka *vozidlo_vlastni* je zde pro identifikaci vozidel, které náleží zákazníkům. Jedná se o vozidla, která jsou přijímána na servis a myčku.

Tab. 20: Vozidlo vlastní (Vlastní zpracování)

| Položka\Tabulka | Vozidlo_vlastni |
|------------------|-----------------|
| ID_vvoz (PK) | Int (not null) |
| Win_cislo | Varchar (20) |
| SPZ | Varchar (10) |
| znacka | Varchar (20) |
| Rok_vyroby | integer |
| barva | Varchar (10) |
| Stav_km | integer |
| ID_zak (FK) | Int |
| ID_typ_vozu (FK) | Int |

Tabulky *objednavka* a *mycka* slouží přijetí vozidla. V tabulce *objednavka* se pouze vyskytuje důvod objednávky, zatímco v tabulce *mycka* způsoby mytí.

Tab. 21: Objednávka (Vlastní zpracování)

| Položka\Tabulka | Objednavka |
|-----------------|----------------|
| Cislo_obj (PK) | Int (not null) |
| ID_zak (FK) | Int |
| ID_vvoz (FK) | Int |
| Duvod_obj | Text |
| Datum | Date |

Tab. 22: Myčka (Vlastní zpracování)

| Položka\Tabulka | Mycka |
|---------------------|----------------|
| Cislo_Dod_list (PK) | Int (not null) |
| ID_zak (FK) | int |
| ID_vvoz (FK) | Int |
| datum | date |
| chemie | bit |
| Spod_mytí | Bit |
| Vysokotlak_bok | Bit |
| Odstran_hmyz | bit |
| Priplatek_ruc_mytí | bit |

Tabulka *material* slouží k evidenci veškerého materiálu či náhradních dílu na skladě.

Tab. 23: Materiál (Vlastní zpracování)

| Položka\Tabulka | Material |
|-----------------|----------------|
| ID_mat (PK) | Int (not null) |
| nazev | Varchar (20) |
| mnozstvi | Integer |
| ID_dod (FK) | Int |
| cena | integer |

Tabulka *servis* rozšiřuje tabulku *objednavka*, zde se zaznamenává čas jednotlivých oprav, popis těchto oprav a kdy byly dokončeny.

Tab. 24: Servis (Vlastní zpracování)

| Položka\Tabulka | Servis |
|-----------------|----------------|
| ID_servis (PK) | Int (not null) |
| ID_sluzba (FK) | Int |

| | |
|-----------------|---------|
| Datum_dokonceni | Date |
| Doba_opravy_el | Integer |
| Doba_opravy_man | integer |
| Popis_oprav | Text |
| Cislo_obj | int |

Tabulka *pouzity_mat* propojuje tabulky *servis* a *material*, což znamená, že je tak možno zjistit jaký materiál byl použit v rámci servisu.

Tab. 25: Použitý materiál (Vlastní zpracování)

| Položka\Tabulka | Pouzity_mat |
|-----------------|----------------|
| ID_mat (PK) | Int (not null) |
| ID_servis (PK) | int |
| Mnozstvi | Integer |

Tabulka *cenik_mycky* je zde jako číselník cen porovnávajících způsoby oplachů a typ vozidla.

Tab. 26: Ceník myčky (Vlastní zpracování)

| Položka\Tabulka | Cenik_mycky |
|---------------------|----------------|
| ID_cena_m (PK) | Int (not null) |
| ID_typ_vozu (FK) | int |
| Cena_chemie | Integer |
| Cena_vysokotlak_bok | Integer |
| Cena_odstran_hmyz | integer |

Tabulky *prace_servis* a *prace_mycka* slouží jako propojení zaměstnance se servisem nebo myčkou a určují tak, který zaměstnanec práci vykonával.

Tab. 27: Práce servis (Vlastní zpracování)

| Položka\Tabulka | Prace_sevis |
|-----------------|----------------|
| ID_servis (PK) | Int (not null) |
| ID_zam (PK) | int |

Tab. 28: Práce myčka (Vlastní zpracování)

| Položka\Tabulka | Prace_mycka |
|---------------------|----------------|
| Cislo_dod_list (PK) | Int (not null) |
| ID_zam (PK) | int |

4.5 Fyzický návrh databáze

Této fázi už je možné na základně předchozího návrhu vytvořit fyzický model, což znamená, že už je možné použít databázi k uchovávání dat. K vytvoření fyzické úrovně jsem použil jazyk SQL a ten byl následně vnesen do programu Microsoft SQL Server 2014 Management studio.

Kód pro vytvoření databáze se nachází v příloze č. 5. V této kapitole pouze znázorním některé stěžejní funkce, které by měli v databázi probíhat.

4.5.1 Trigger pro vyskladnění vozidla

Tento skript představuje vytvoření triggeru, který se spustí, jakmile se se vytvoří nový záznam v tabulce *smlouva_prod*, jenž obsahuje informace i prodávaném vozidle. To znamená, že na základě ID vozidla obsaženém ve vytvořené smlouvě, trigger označí vozidlo jako vyskladněné a přenesení jeho údaje do tabulky *prodana_vozidla*.

```
create trigger vyskladnit on smlouva_prod
for insert
as
Begin
update sklad_vozidel
set skladem=0
from sklad_vozidel join inserted on sklad_vozidel.ID_svoz = inserted.ID_svoz
insert into prodana_vozidla
(win_cislo,znacka,rok_vyroby,barva,stav_km,vybava,CZK_bez_DPH,EUR_bez_DPH)
select win_cislo,znacka,rok_vyroby,barva,stav_km,vybava,CZK_bez_DPH,EUR_bez_DPH
from sklad_vozidel where skladem = 0
end
```

4.5.2 Procedury na vystavení faktury

Jelikož tuto databázi zaměřuji především na vystavení faktur, tak je zde popíšu procedury k tomu určené.

Nejprve se jedná o proceduru *vystavit_fakturu_p*, která má za úkol vystavit fakturu v oblasti prodeje, což znamená, že se vloží údaje a procedura na testuje, v jaké měně chce zákazník dané vozidlo zaplati a jaké DPH se vztahuje k dokladu a na základě těchto informací vystaví doklad se splatností 30 dní ode dne vystavení dokladu.

```

create procedure vystavit_fakturu_p
@cislof int,
@dph int,
@forma int,
@mena int,
@zak int,
@zam int,
@voz int
as
begin
declare @cenab integer,
        @cenas integer,
        @dphh integer
if @dph = 1 select @dphh = hodnota from DPH where @dph = ID_dph
else select @dphh = hodnota from DPH where @dph = ID_dph + 1
if @mena = 1
    select @cenab = CZK_bez_DPH, @cenas =
(CZK_bez_DPH+(CZK_bez_DPH*@dphh/100)) from sklad_vozidel where @voz = ID_svoz
else
    select @cenab = EUR_bez_DPH, @cenas =
(EUR_bez_DPH+(EUR_bez_DPH*@dphh/100)) from sklad_vozidel where @voz = ID_svoz

insert into faktura (cislo_fak,ID_dph,ID_mena,ID_forma,ID_zak,ID_zam, var_symbol,
cena_bez_dph, cena_s_dph, datum_vystaveni,datum_splatnosti,oblast_vystaveni)
values (@cislof, @dph, @mena, @forma, @zak, @zam, @cislof, @cenab, @cenas,
GETDATE(),GETDATE()+30,'prodej')
end
go

```

Následující procedura *vystavit_fakturu_m* se vztahuje na vystavení faktury za použití mycí linky, kde se opět vloží potřebné údaje z dodacího listu a podle toho pak procedura vyhodnocuje jaké programy, zde byli použity, které pak náležitě cenové zhodnotí a vystaví doklad na stejném principu jako předcházející procedura.

```

create procedure vystavit_fakturu_m
@zak int,
@zam int,
@fak int,
@dph int,
@forma int,
@myc int,
@voz int
as
begin
declare @cenab integer,
        @cenas integer,
        @dphh integer,
        @normal int,
        @chem int,
        @spod int,
        @vysok int,

```



```

        @hmyz int,
        @prip int,
        @typ int,
        @cenam integer

set @cenab=0
set @normal= (select normal from mycka where cislo_dod_list=@myc)
set @chem= (select chemie from mycka where cislo_dod_list=@myc)
set @spod= (select spod_myti from mycka where cislo_dod_list=@myc)
set @vysok= (select vysokotlak_bok from mycka where cislo_dod_list=@myc)
set @hmyz= (select odstran_hmyz from mycka where cislo_dod_list=@myc)
set @prip= (select priplatek_ruc_myti from mycka where cislo_dod_list=@myc)
set @typ= (select ID_typ_vozu from vozidlo_vlastni where @voz=ID_vvoz)

if @dph = 1 select @dphh = hodnota from DPH where @dph = ID_dph
    else select @dphh = hodnota from DPH where @dph = ID_dph + 1

if @normal=1 select @cenam=cena_normal from cenik_mycky where @typ =ID_typ_vozu
    set @cenab=@cenam
if @chem=1 select @cenab=cena_chemie from cenik_mycky where @typ =ID_typ_vozu
    set @cenab=@cenab+ @cenam
if @spod=1 select @cenam=cena_spod_myti from cenik_mycky where @typ =ID_typ_vozu
    set @cenab=@cenab+ @cenam
if @vysok=1 select @cenam=cena_vysokotlak_bok from cenik_mycky where @typ
=ID_typ_vozu
    set @cenab=@cenab+ @cenam
if @hmyz=1 select @cenam=cena_odstran_hmyz from cenik_mycky where @typ
=ID_typ_vozu
    set @cenab=@cenab+ @cenam
if @prip=1 select @cenam=250 from cenik_mycky where @typ =ID_typ_vozu
    set @cenab=@cenab+ @cenam
    set @cenas=@cenab+ @cenab*@dphh/100

insert into faktura (cislo_fak,ID_dph,ID_mena,ID_forma,ID_zak,ID_zam, var_symbol,
cena_bez_dph, cena_s_dph, datum_vystaveni,datum_splatnosti,oblast_vystaveni)
values (@fak,@dph, 1, @forma, @zak, @zam, @fak, @cenab, @cenas,
GETDATE(),GETDATE()+30,'myci linka')

end

```

Proceduru na vystavení dokladu za využití služeb servisního centra zde už nezahrnuji, protože by se v zásadě jednalo o stejný princip.

4.6 Řešení bezpečnosti

V této podkapitole je třeba zaměřit se také na bezpečnost databáze jak z hlediska možného napadení či nežádoucího přístupu cizích osob, tak z hlediska ztráty potřebných dat. Následující možnosti by měly tyto rizika snížit.

4.6.1 Rozdělení přístupových údajů

Aby bylo možné se pro zaměstnance dostat do databáze, přiřadil jsem každému z nich údaje, za pomoci kterých získají přístup do databáze. Jedná se o údaje login a heslo, které se přidělují při vložení, ale to nemusí být podmínka. Zaměstnanci, kteří nebudou důkladně seznámeni s chodem a obsluhou databáze, dostanou přístupové údaje až po proškolení. Tyto údaje může přidělovat pouze vedoucí pracovník, který je k tomu způsobilý. Na obr. 13 jsou zobrazeny údaje, které již byly zaměstnancům přiděleny.

| | jmeno | příjmení | logi | heslo |
|---|------------|-----------|----------|---------|
| 1 | Daniel | Dvořák | dvorak11 | heslo11 |
| 2 | Tomáš | Herman | herman11 | heslo22 |
| 3 | Stanislava | Dvořáková | dvorak33 | heslo33 |
| 4 | Josef | Kalina | kalina44 | heslo44 |
| 5 | Aleš | Procházka | dvorak55 | heslo55 |
| 6 | Marcela | Jurová | jurova66 | heslo66 |

Obr. 13: Zaměstnanci, jimž byly přiděleny přístupové údaje (Vlastní zpracování)

4.6.2 Rozdělení zaměstnaneckých rolí

Protože se jedná o systém, který pokrývá víc oblastí, rozhodl jsem se rozdělit jej z hlediska bezpečnosti do sekcí, do nichž se budou zaměstnanci moci podle své pozice ve firmě. Například zaměstnanec ze servisu nebude moci vstoupit do sekce prodeje a naopak. Jedná se tedy o sekce prodeje, správy osob, servisu a mycí linky. Jednotlivé sekce je možno vidět v přílohách 1 až 4. Rozdělení do sekcí jsem provedl pomocí modulu Security.

Aby toto rozdělení k něčemu bylo, je nutné rozdělit mezi jednotlivé uživatele rozdělit role, díky nimž se zamezí ztrátě či chybovosti dat. Pomocí následujících ukázkových příkazů jsem vytvořil role *technik* a *vedouci_servisu*, k nimž jsem přiřadil práva, která budou moci vykonávat v rámci oblasti servisu.

```
create role technik
create role vedouci_servisu

grant insert,select,update on schema:: servis to technik
grant insert,select,update,execute,delete on schema::servis to vedouci_servisu
grant insert,select,update,execute,delete on schema::faktury to vedouci_servisu
```

Z příkazu se dá pochopit, že role *technik* nemá taková práva jako role *vedouci_servisu*, to znamená, že pracovník nižší pozice bude způsobilý pouze ke vkládání a aktualizaci dat v rámci servisu. Pokud by došlo k potřebě mazat nějaké údaje či vystavit fakturu, tak k tomu už je způsobilý vedoucí servisu. Tento typ zabezpečení je pak modifikován i na ostatní oblasti.

Předchozí možnosti zabezpečení řeší bezpečnost pouze v rámci databáze, aby byla bezpečnost na vyšší úrovni, je nutné řešit taky otázku zabezpečení sítě, z čehož vyplívají další možnosti opatření. Jelikož tato práce nemá cíl řešit bezpečnost na síťové úrovni, nebudu se tímto tématem dále zabývat.

4.7 Budoucí uživatelské rozhraní

Firma se rozhodla investovat do vývoje aplikace, která bude propojena s navrhovanou databází. Tato aplikace by pak měla ulehčit veškeré administrativní procesy. Než bude toto řešení zhotoveno, navrhoval bych dočasné pořízení programu Microsoft SQL Sever 2014 Standard edition.

Toto řešení by mohlo zatím seznámit zaměstnance s databází a navíc celá firma má platformu na bázi Microsoft Windows, což je jediné známe uživatelské rozhraní pro zaměstnance. Cena toho systému je vymezena na 27 555 Kč.

4.8 Předpokládané přínosy návrhu pro podnik

Jelikož cílem této práce bylo vytvořit pouze návrh databáze, který by dokázal pokrýt celou funkcionalitu podniku, je tedy těžké určit možné přínosy z tohoto návrhu. Důvodem je aplikační rozhraní, jenž by dokázal zaměstnancům usnadnit obsluhu této databáze.

Z finanční stránky je možno odhadnout v jakém cenovém rozmezí by se zhruba mohl celý vývoj takového rozhraní pohybovat. Ačkoliv neznám přesnou hodnotu odhaduji, že cena takovéto aplikace by byla v rozmezí 15 000 až 30 000 Kč. K tomu připočtu cenu dočasného pořízení programu Microsoft SQL Server 2014 Standard edition 27 555 Kč plus případné školení zaměstnanců na obsluhu v tomto uživatelském prostředí. Cena se tedy ve finále pohybuje na rozmezí 50 000 až 60 000 Kč.

Na základě uvedeného návrhu je možno říct, že přínosy budou po všech stránkách z hlediska administrativy. Zaměstnanci se budou moci dostat vždy aktuálním informací, ať už se jedná o kontaktní údaje zákazníku, proběhlých obchodech či zakázkách právě probíhajících na servisu. To znamená, že se tyto informace budou soustředěny na podnikovém serveru. Vzhledem tomu, že některá data byla uchovávána pouze papírovými záznamy, jako byli podklady k fakturám, nebo exelovými tabulkami, které se vyskytovaly na více místech, odpadne tak složité přeposílání jednotlivých souborů mezi zaměstnanci.

Následující body by měly objasnit veškeré možné přínosy tohoto návrhu.

- Získávání aktuálních údajů z databáze,
- komplexní přehled vozidel určených k prodeji a již prodaných vozidel,
- možnost nahlížení na aktuální a proběhlé zakázky v servisu, v případě potíží snadné dohledání zaměstnanců, kteří zakázky vykonali,
- vystavování faktur v celé oblasti podniku, následně možná modifikace či stornování fakturačních dokladů,
- možnost kontroly o zaplacených dokladech,
- snadnější dohledávání kupních smluv, zda jsou platné atd.

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout databázi, jenž by pokryla celou oblast podnikání firmy Dvořák Trucks s.r.o. a zjednodušit tak dosavadní práci zaměstnanců. To zahrnuje jednoduchou evidenci kontaktních údajů všech osob (zákazníku, zaměstnanců a dodavatelů), orientaci ve všech svých objednávkách či zakázkách. Dále všechny uskutečněné prodeje, které se vážou na kupní smlouvu a také prodaná vozidla, podle kterých firma je schopna rozhodovat, která jsou nejvíce prodávána. V neposlední řadě by zde měla být oblast na vystavení fakturačních dokladů, za prodej firemních služeb, kde bude možno vidět informace o jednotlivých dokladech a zda již byly zaplacené.

Na základě požadavků na databázi a současné situace v podniku jsem byl schopen vytvořit návrh databáze, který splňuje jak otázku z hlediska funkcionality tak i z hlediska bezpečnosti na aplikační úrovni. Cíl bakalářské práce je tedy splněn, databáze pokrývá všechny oblasti, které byly označeny požadavcích a jedná se o oblasti správy osob, prodeje, servisu a mycí linky a vystavování faktur.

Jelikož se jedná pouze o návrh, je nutné, aby dalším krokem byla realizace samotného aplikačního rozhraní, které bylo propojeno s navrhovanou databází a umožňovalo tak zaměstnancům jednoduché používání této databáze.

SEZNAM POUŽITÝCH ZDROJŮ

- (1) SKLENÁK, Vilém. *Data, informace, znalosti a Internet*. Praha: C.H. Beck, 2001. C.H. Beck pro praxi. ISBN 80-717-9409-0.
- (2) BASL, Josef a Roman BLAŽÍČEK. *Podnikové informační systémy: podnik v informační společnosti*. 3., aktualiz. a dopl. vyd. Praha: Grada, 2012. *Management v informační společnosti*. ISBN 978-80-247-4307-3.
- (3) CONOLLY, Thomas, Carolyn E. BEGG a Richard HOLOWCZAK. *Mistrovství - databáze: profesionální průvodce tvorbou efektivních databází*. Brno: Computer Press, 2009. ISBN 978-80-251-2328-7.
- (4) KŘÍŽ, Jiří a Petr DOSTÁL. *Databázové systémy*. Brno: Akademické nakladatelství CERM, 2005. ISBN 80-214-3064-8.
- (4) KOCH, Miloš a Bernard NEUWIRTH. *Datové a funkční modelování*. Vyd. 4., rozš. Brno: Akademické nakladatelství CERM, 2010. ISBN 978-80-214-4125-5.
- (5) KROENKE, David a David J. AUER. *Databáze*. Brno: Computer Press, 2015. ISBN 978-80-251-4352-0.
- (6) BERRINGTON, James. *Databases. Anaesthesia & Intensive Care Medicine*. 2014, (14), 61. ISSN 1472-0299. Dostupné z: <http://www.sciencedirect.com.ezproxy.lib.vutbr.cz/science/article/pii/S1472029913003147>
- (7) LACKO, L. *1001 tipů a triků pro SQL*. 1. vyd. Brno: Computer Press, 2011. 416s. ISBN 978-80-251-3010-0.
- (8) OPPEL, Andrew J. *SQL bez předchozích znalostí: [průvodce pro samouky]*. Brno: Computer Press, 2008. ISBN 978-80-251-1707-1.
- (9) POKORNÝ, Jaroslav a Michal VALENTA. *Databázové systémy*. Praha: České vysoké učení technické v Praze, 2013, v, 265 stran. ISBN 978-80-01-05212-9.

(10) LACKO, Ľuboslav a David J. AUER. Mistrovství v SQL Server 2012: [kompletní průvodce databázového experta]. Brno: Computer Press, 2013. ISBN 978-80-251-3773-4.

(11) GRASSEOVÁ, Monika, Radek DUBEC a David ŘEHÁK. *Analýza v rukou manažera: 33 nejpoužívanějších metod strategického řízení*. Brno: Computer Press, 2010. ISBN 978-80-251-2621-9.

SEZNAM POUŽITÝCH ZKRATEK A SYMBOLŮ

| | |
|----------|---|
| SQL | Structured query language |
| DBMS | Database management server |
| NF | Normální forma |
| SWOT | Strengths, weaknesses, opportunities, threats |
| ER model | Entity-relationship model |
| IS | Informační systém |
| PK | Primary key |
| FK | Foreign key |
| ID | Identify |

SEZNAM OBRÁZKŮ

| | |
|---|----|
| Obr. 1: Data, poznatky, informace..... | 11 |
| Obr. 2: Komunikace uživatelů s databází | 13 |
| Obr. 3: Organizační struktura | 25 |
| Obr. 4: Příjmová objednávka na servis | 31 |
| Obr. 5: Příjmoví dodací list k mycí lince..... | 32 |
| Obr. 6: Vývojový diagram procesu evidence zákazníka | 37 |
| Obr. 7: Vývojový diagram pro tvorbu kupní smlouvy | 38 |
| Obr. 8: Vývojový diagram procesu pro příjem k servisu | 39 |
| Obr. 9: Vývojový diagram sekce faktur..... | 40 |
| Obr. 10: Vývojový diagram pro vytvoření smluvního vztahu..... | 41 |
| Obr. 11: ER-diagram konceptuální úrovně..... | 42 |
| Obr. 12: Diagram logické úrovně | 43 |

SEZNAM TABULEK

| | |
|---|----|
| Tab. 1: Datové typy SQL..... | 22 |
| Tab. 2: Pozice | 44 |
| Tab. 3: Zaměstnanec | 44 |
| Tab. 4: Zákazník | 45 |
| Tab. 5: Dodavatel..... | 45 |
| Tab. 6: Forma úhrady..... | 45 |
| Tab. 7: Měna | 46 |
| Tab. 8: DPH | 46 |
| Tab. 9: Faktura..... | 46 |
| Tab. 10: Faktura prodeje..... | 46 |
| Tab. 11: Faktura servis..... | 47 |
| Tab. 12: Faktura myčka | 47 |
| Tab. 13: Smlouva pro čerpání slev | 47 |
| Tab. 14: Použití slevy | 47 |
| Tab. 15: Typ vozu..... | 48 |
| Tab. 16: Sklad vozidel | 48 |
| Tab. 17: Prodaná vozidla | 48 |
| Tab. 18: Smlouva prodeje..... | 49 |
| Tab. 19: Služba | 49 |

| | |
|--------------------------------|----|
| Tab. 20: Vozidlo vlastní..... | 49 |
| Tab. 21: Objednávka..... | 50 |
| Tab. 22: Myčka | 50 |
| Tab. 23: Materiál..... | 50 |
| Tab. 24: Servis | 50 |
| Tab. 25: Použitý materiál..... | 51 |
| Tab. 26: Ceník myčky..... | 51 |
| Tab. 27: Práce servis..... | 51 |
| Tab. 28: Práce myčka..... | 51 |

SEZNAM PŘÍLOH

Příloha 1: Diagram sekce osob

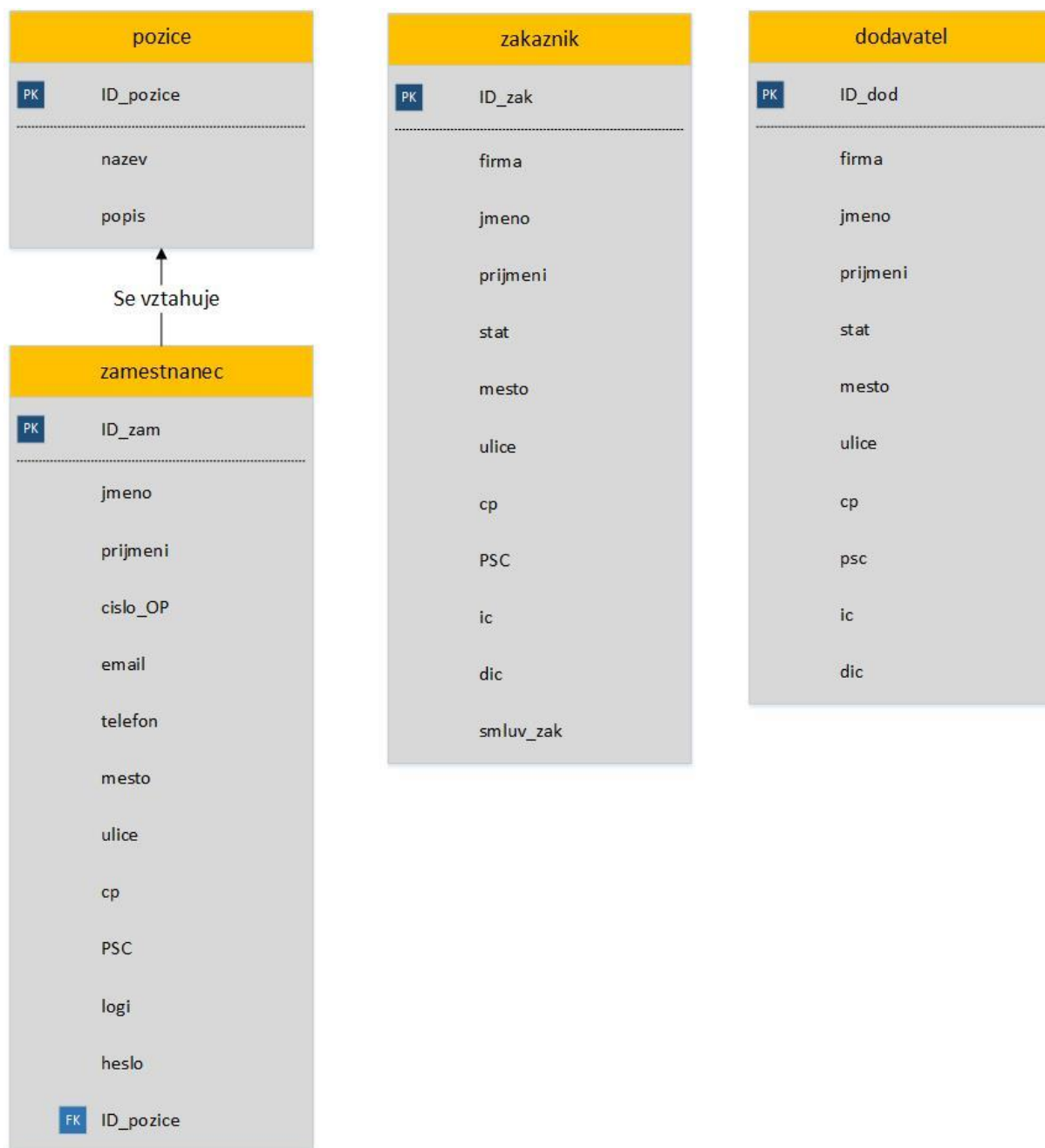
Příloha 2: Diagram sekce prodeje

Příloha 3: Diagram sekce faktur

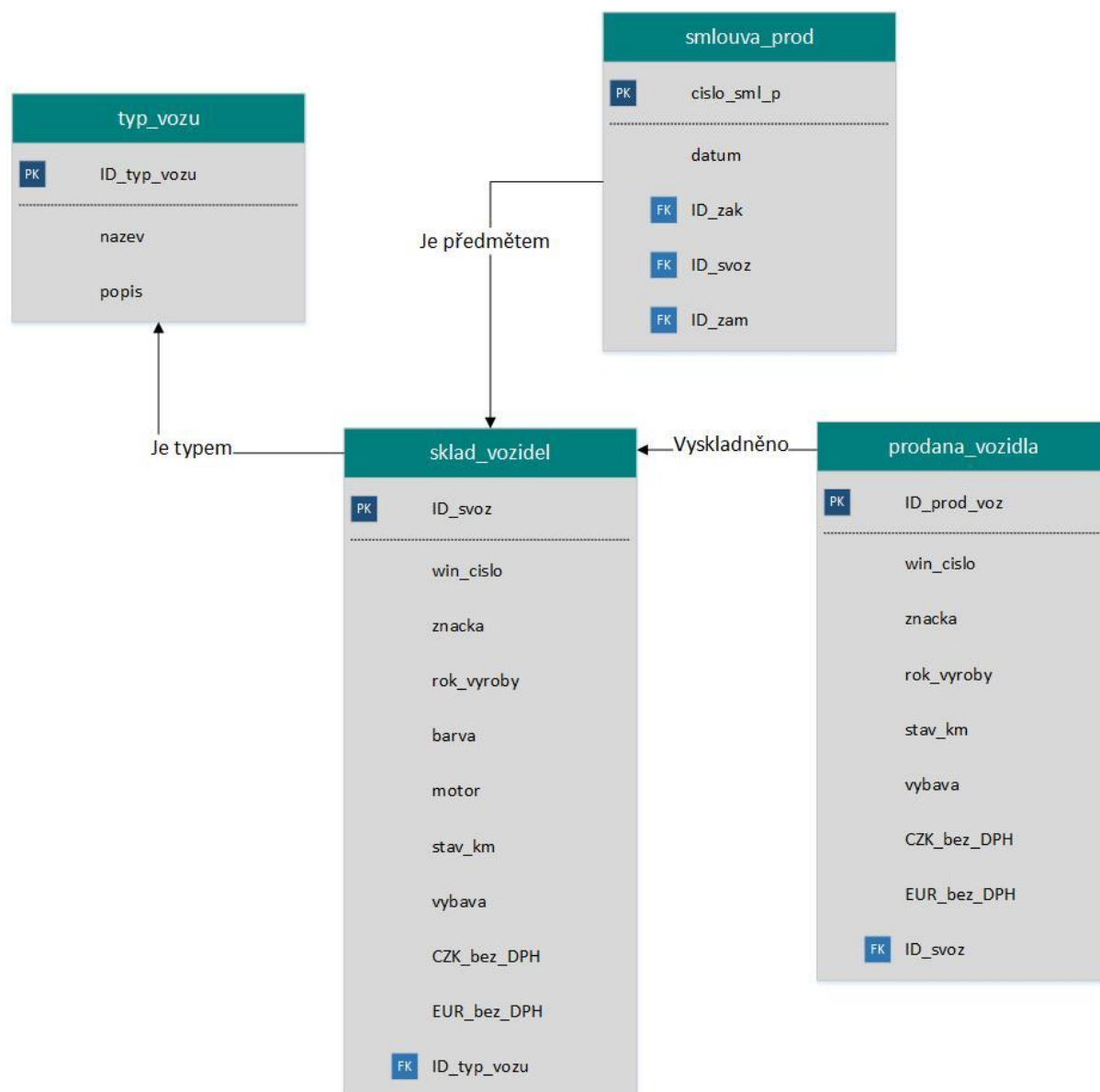
Příloha 4: Diagram sekce servisu a mycí linky

Příloha 5: Kód pro vytvoření databáze

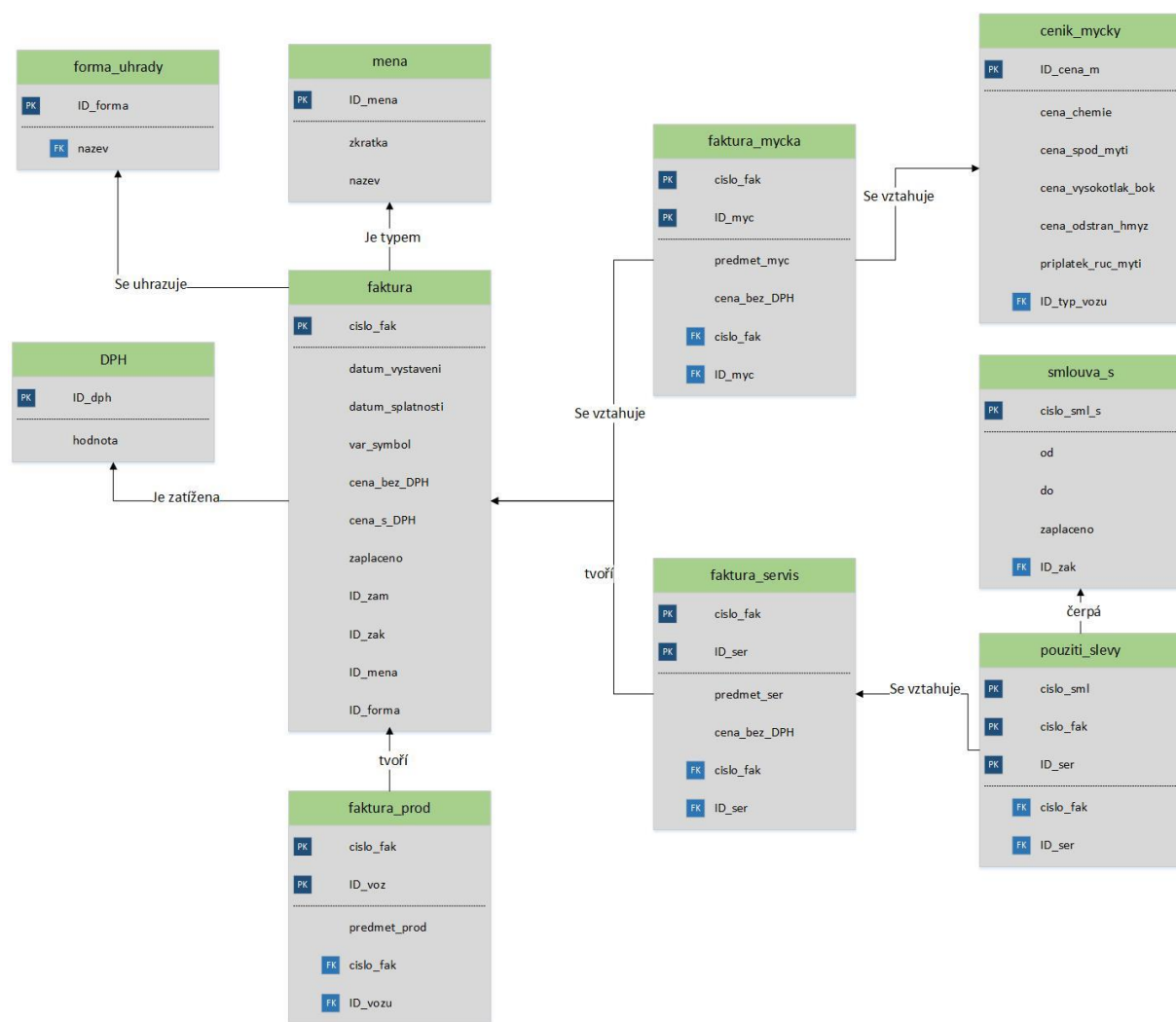
Příloha 1: Diagram sekce osob



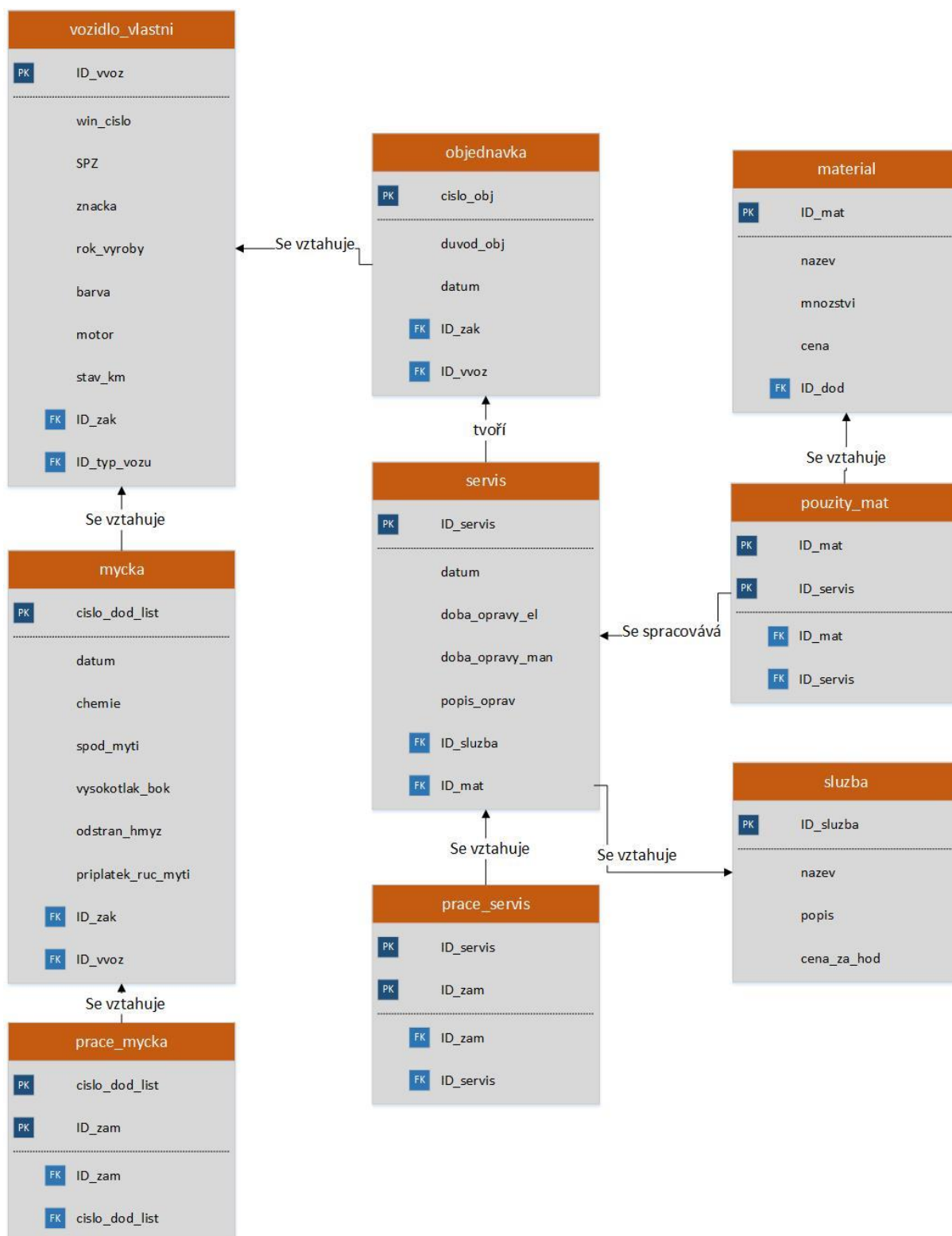
Příloha 2: Diagram sekce prodeje



Příloha 3: Diagram sekce faktur



Příloha 4: Diagram sekce servisu a mycí linky



Příloha 5: Kód pro vytvoření databáze

```
Create table pozice (  
ID_pozice int identity (1,1),  
nazev varchar (20) not null,  
zkratka varchar (5) not null,  
popis text,  
constraint PK_tb_poz primary key (ID_pozice))  
go
```

```
Create table zamestnanec (  
ID_zam int identity (1,1),  
jmeno varchar(30) not null,  
prijmeni varchar (30) not null,  
cislo_OP varchar (9) not null,  
email varchar(60),  
telefon varchar(13),  
mesto varchar (20) not null,  
ulice varchar (20) not null,  
cp integer not null,  
PSC integer not null,  
logi varchar (20),  
heslo varchar (20),  
ID_pozice int,  
constraint PK_tb_zam primary key (ID_zam),  
constraint FK_pozice foreign key (ID_pozice) references pozice (ID_pozice))  
go
```

```
create table zakaznik (  
ID_zak int identity (1,1),  
firma varchar (50),
```

```

jmeno varchar (30) not null,
prijmeni varchar (30) not null,
stat varchar (20) not null,
mesto varchar (20) not null,
telefon varchar (13) not null,
ulice varchar (20) not null,
cp integer not null,
psc integer not null,
ic integer,
dic integer,
smluv_zak bit,
constraint PK_tb_zak primary key (ID_zak))
go

```

```

create table dodavatel (
ID_dod int identity (1,1),
firma varchar (50) not null,
jmeno varchar (30) not null,
prijmeni varchar (30) not null,
stat varchar (20) not null,
telefon varchar (13) not null,
mesto varchar (20) not null,
ulice varchar (20) not null,
cp integer not null,
psc integer not null,
ic integer,
dic integer,
constraint PK_tb_dod primary key (ID_dod))

create table typ_vozu (
ID_typ_vozu int identity (1,1),

```

```

nazev varchar (20) not null,

popis text not null,

constraint PK_tb_typvz primary key (ID_typ_vozu))

go

```

```

create table sklad_vozidel (

ID_svoz int identity (1,1),

win_cislo varchar (20) not null,

znacka varchar (20) not null,

rok_vyroby integer not null,

barva varchar (10) not null,

motor varchar (15),

stav_km integer not null,

vybava text not null,

CZK_bez_DPH integer not null,

EUR_bez_DPH integer not null,

skladem bit,

ID_typ_vozu int,

ID_dod int,

constraint PK_tb_sklad_voz primary key (ID_svoz),

constraint FK_tb_typv foreign key (ID_typ_vozu) references typ_vozu (ID_typ_vozu),

constraint FK_tb_sdod foreign key (ID_dod) references dodavatel (ID_dod))

go

```

```

create table prodana_vozidla (

ID_prod_voz int identity (1,1),

ID_typ_vozu int,

win_cislo varchar (20),

znacka varchar (20),

rok_vyroby integer,

```

```

barva varchar (10),
motor varchar (15),
stav_km integer,
vybava text,
CZK_bez_DPH integer,
EUR_bez_DPH integer,
constraint PK_tb_prod_voz primary key (ID_prod_voz),
constraint FK_tb_voz foreign key (ID_typ_vozu) references typ_vozu (ID_typ_vozu))
go

```

```

create table mena (
ID_mena int identity (1,1),
zkratka varchar (4) not null,
nazev varchar (20) not null,
constraint PK_tb_mena primary key (ID_mena))
go

```

```

create table forma_uhrady (
ID_forma int identity (1,1),
nazev varchar (20) not null,
constraint PK_tb_forma primary key (ID_forma))
go

```

```

create table DPH (
ID_dph int identity (1,1),
hodnota integer not null,
constraint PK_tb_dph primary key (ID_dph))
go

```

```

create table faktura (
cislo_fak int not null,
oblast_vystaveni varchar(10) not null,
datum_vystaveni date not null,
datum_splatnosti date not null,
var_symbol integer not null,
cena_bez_DPH integer not null,
cena_s_DPH integer not null,
ID_zam int not null,
ID_zak int not null,
ID_mena int not null,
ID_forma int not null,
ID_dph int not null,
zaplaceno bit,

constraint PK_tb_fak primary key (cislo_fak),
constraint FK_tb_zam foreign key (ID_zam) references zamestnanec (ID_zam),
constraint FK_tb_zak foreign key (ID_zak) references zakaznik (ID_zak),
constraint FK_tb_mena foreign key (ID_mena) references mena (ID_mena),
constraint FK_tb_forma foreign key (ID_forma) references forma_uhrady (ID_forma),
constraint FK_tb_dph foreign key (ID_dph) references dph (ID_dph))

go

```

```

create table faktura_prod (
cislo_fak int not null,
ID_voz int not null,
predmet_prod varchar (20),

constraint PK_tb_fakprod primary key (cislo_fak, ID_voz),
constraint FK_tb_cislofp foreign key (cislo_fak) references faktura (cislo_fak),
constraint FK_tb_vozp foreign key (ID_voz) references sklad_vozidel (ID_svoz))

go

```

```

create table smlouva_s (
cislo_sml_s int identity (1,1),
od date not null,
do date not null,
ID_zak int not null,
zaplaceno bit,
constraint PK_tb_sml_s primary key (cislo_sml_s),
constraint FK_tb_smls_zak foreign key (ID_zak) references zakaznik (ID_zak))
go

```

```

create table smlouva_prod (
cislo_sml_p int identity (1,1),
popis text,
datum date not null,
ID_zak int not null,
ID_zam int not null,
ID_svoz int not null,
constraint PK_tb_sml_p primary key (cislo_sml_p),
constraint FK_tb_smlp_zak foreign key (ID_zak) references zakaznik (ID_zak),
constraint FK_tb_smlp_svoz foreign key (ID_svoz) references sklad_vozidel (ID_svoz),
constraint FK_tb_smlp_zam foreign key (ID_zam) references zamestnanec (ID_zam))
go

```

```

create table sluzba (
ID_sluzba int identity (1,1),
nazev varchar (20) not null,
popis text,
cena_za_hod integer not null,

```

```

constraint PK_tb_sluz primary key (ID_sluzba))

create table vozidlo_vlastni (
ID_vvoz int identity (1,1),
win_cislo varchar (20) not null,
SPZ varchar (10) not null,
znacka varchar (20) not null,
rok_vyroby integer not null,
barva varchar (10),
motor varchar (15),
stav_km integer not null,
ID_zak int not null,
ID_typ_vozu int not null,
constraint PK_tb_vlast_voz primary key (ID_vvoz),
constraint FK_tb_voz_zak foreign key (ID_zak) references zakaznik (ID_zak),
constraint FK_tb_typ_vozu foreign key (ID_typ_vozu) references typ_vozu (ID_typ_vozu))
go

```

```

create table objednavka (
cislo_obj int identity (1,1),
ID_zak int not null,
ID_vvoz int not null,
duvod_obj text not null,
datum date not null,
constraint PK_tb_objednavka primary key (cislo_obj),
constraint FK_tb_obj_zak foreign key (ID_zak) references zakaznik (ID_zak),
constraint FK_tb_vlast_voz foreign key (ID_vvoz) references vozidlo_vlastni (ID_vvoz))
go

```

```

create table material (
ID_mat int identity (1,1),
nazev varchar(20) not null,
mnozstvi integer not null,
ID_dod int not null,
cena integer not null,
constraint PK_tb_mat primary key (ID_mat),
constraint FK_tb_mat_dod foreign key (ID_dod) references dodavatel (ID_dod))
go

```

```

create table servis (
ID_servis int identity (1,1),
ID_sluzba int not null,
datum date not null,
doba_opravy_el integer not null,
doba_opravy_man integer not null,
popis_oprav text,
cislo_obj int not null,
constraint PK_tb_servis primary key (ID_servis),
constraint FK_tb_sluzba foreign key (ID_sluzba) references sluzba (ID_sluzba),
constraint FK_tb_mat foreign key (ID_mat) references material (ID_mat),
constraint FK_tbobj foreign key (cislo_obj) references objednavka (cislo_obj))
go

```

```

create table pouzity_mat (
ID_mat int not null,
ID_servis int not null,
mnozstvi integer not null,
constraint PK_tb_pmat primary key (ID_mat,ID_servis),
constraint FK_tb_pm_mat foreign key (ID_mat) references material (ID_mat),
constraint FK_tb_pm_ser foreign key (ID_servis) references servis (ID_servis))

```



```

create table cenik_mycky (
ID_cena_m int identity (1,1),
ID_typ_vozu int not null,
cena_normal integer not null,
cena_chemie integer not null,
cena_spod_myti integer not null,
cena_vysokotlak_bok integer not null,
cena_odstran_hmyz integer not null,
constraint PK_tb_cm primary key (ID_cena_m),
constraint FK_tb_cm_typv foreign key (ID_typ_vozu) references typ_vozu (ID_typ_vozu))
go

```

```

create table mycka (
cislo_dod_list int identity (1,1),
ID_zak int not null,
ID_vvoz int not null,
datum date not null,
normal bit,
chemie bit,
spod_myti bit,
vysokotlak_bok bit,
odstran_hmyz bit,
priplatek_ruc_myti bit,
constraint PK_tb_mycka primary key (cislo_dod_list),
constraint FK_tb_zakm foreign key (ID_zak) references zakaznik (ID_zak),
constraint FK_tb_vvozm foreign key (ID_vvoz) references vozidlo_vlastni (ID_vvoz))
go

```

```

create table prace_servis (
ID_servis int not null,
ID_zam int not null,
constraint PK_tb_praces primary key (ID_servis,ID_zam),
constraint FK_tb_servis_zam foreign key (ID_zam) references zamestnanec (ID_zam),
constraint Fk_tb_servis foreign key (ID_servis) references servis (ID_servis))
go

```

```

create table prace_mycka (
cislo_dod_list int not null,
ID_zam int not null,
constraint PK_tb_pracem primary key (cislo_dod_list,ID_zam),
constraint FK_tb_mycka_zam foreign key (ID_zam) references zamestnanec (ID_zam),
constraint Fk_tb_mycka foreign key (cislo_dod_list) references mycka (cislo_dod_list))
go

```

```

create table faktura_servis (
cislo_fak int identity (1,1),
ID_ser int not null,
predmet_ser varchar (20) not null,
cena_bez_DPH integer not null,
constraint PK_tb_fakser primary key (cislo_fak,ID_ser),
constraint FK_tb_cislofs foreign key (cislo_fak) references faktura (cislo_fak),
constraint FK_tb_ser foreign key (ID_ser) references servis (ID_servis))
go

```

```

create table faktura_mycka (
cislo_fak int identity (1,1),
ID_myc int not null,

```

```

predmet_myc varchar (20),

cena_bez_DPH integer not null,

constraint PK_tb_fakmyc primary key (cislo_fak,ID_myc),

constraint FK_tb_cislofm foreign key (cislo_fak) references faktura (cislo_fak),

constraint FK_tb_myc foreign key (ID_myc) references mycka (cislo_dod_list)

go


create table pouziti_slevy (

cislo_sml_s int not null,

cislo_fak int not null,

ID_ser int not null,

constraint PK_tb_sleva primary key (cislo_sml_s,cislo_fak,ID_ser),

constraint FK_tb_slevaF foreign key (cislo_fak,ID_ser) references faktura_servis
(cislo_fak,ID_ser),

constraint FK_tb_slevas foreign key (cislo_sml_s) references smlouva_s (cislo_sml_s))

go

```